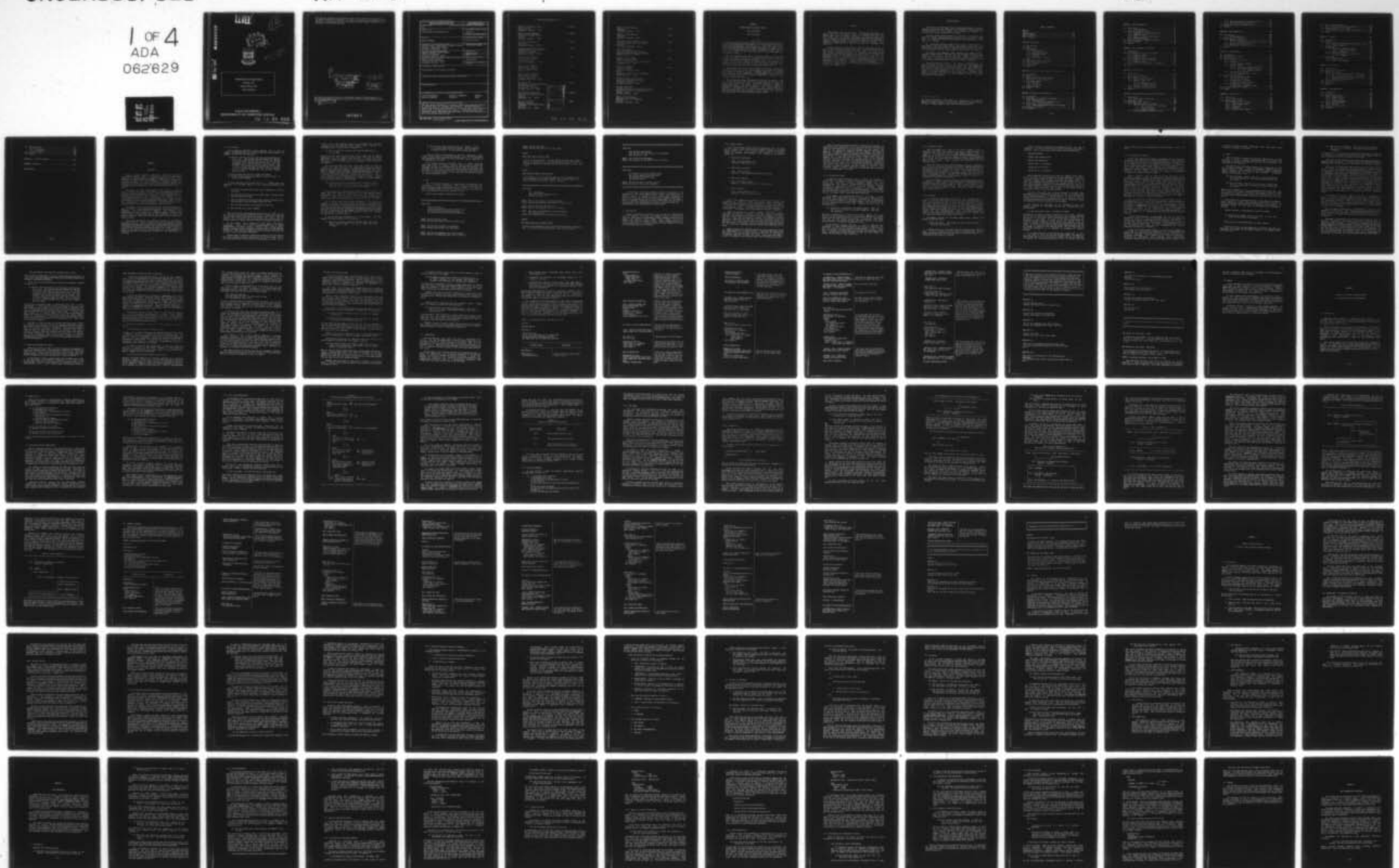
AD-A062 629 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 5/7
UNDERSTANDING GOAL-BASED STORIES.(U)
SEP 78 R WILENSKY N00014-75-C-1111
UNCLASSIFIED RR-140 NL

1 OF 4
ADA
062629



LUX ET VERITAS

UNDERSTANDING GOAL-BASED STORIES

September 1978

Research Report #140

Robert Wilensky

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

This work was presented to the Graduate School of Yale University in candidacy for the degree of Doctor of Philosophy. The author is presently in the Computer Science Division of the Department of Electircal Engineering and Computer Science at the University of California, Berkeley.

6

UNDERSTANDING GOAL-BASED STORIES.

11 Sep 78

September 1978

14 RR-

9 Research Report 140

10 Robert Wilensky

12 330p.

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored in part by the Office of Naval Research under contract N00014-75-C-1111

15

407051

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>#140 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>Understanding Goal-Based Stories | | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Robert Wilensky | | 8. CONTRACT OR GRANT NUMBER(s)<br>N00014-75-C-1111. |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Yale University - Department of Computer Science<br>Artificial Intelligence Project<br>New Haven, Connecticut 06520 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Advanced Research Projects Agency<br>1400 Wilson Boulevard<br>Arlington, Virginia 22209 | | 12. REPORT DATE<br>September 1978 |
| | | 13. NUMBER OF PAGES<br>317 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)<br>Office of Naval Research<br>Information Systems Program<br>Arlington, Virginia 22217 | | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

| | | |
|---|---|---|
| story understanding | conceptual dependency | planning |
| question answering | inference | goals |

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

PAM (Plan Applier Mechanism) is a computer program that understands stories by reasoning about the situations they reference. PAM reads stories in English, and produces representations for the stories that include the inferences needed to connect each story's events. To demonstrate that it has understood a story, PAM answers questions about the story and expresses the story from several points of view.

-- OFFICIAL DISTIRUBTION LIST --

| | |
|---|---|
| Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 12 copies |
| Office of Naval Research<br>Information Systems Program<br>Code 437<br>Arlington, Virginia 22217 | 2 copies |
| Office of Naval Research<br>Code 102IP<br>Arlington, Virginia 22217 | 6 copies |
| Advanced Research Projects Agency<br>Cybernetics Technology Office<br>1400 Wilson Boulevard<br>Arlington, Virginia 22209 | 3 copies |
| Office of Naval Research<br>Branch Office - Boston<br>495 Summer Street<br>Boston, Massachusetts 02210 | 1 copy |
| Office of Naval Research<br>Branch Office - Chicago<br>536 South Clark Street<br>Chicago, Illinois 60615 | 1 copy |
| Office of Naval Research<br>Branch Office - Pasadena<br>1030 East Green Street<br>Pasadena, California 91106 | 1 copy |
| Mr. Steven Wong<br>Administrative Contracting Officer<br>New York Area Office<br>715 Broadway - 5th Floor<br>New York, New York 10003 | 1 copy |
| Naval Research Laboratory<br>Technical Information Division<br>Code 2627<br>Washington, D.C. 20375 | 6 copies |
| Dr. A.L. Slafkosky<br>Scientific Advisor<br>Commandant of the Marine Corps<br>Code RD-1<br>Washington, D.C. 20380 | 1 copy |

Office of Naval Research — 1 copy
Code 455
Arlington, Virginia 22217

Office of Naval Research — 1 copy
Code 458
Arlington, Virginia 22217

Naval Electronics Laboratory Center — 1 copy
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E.H. Gleissner — 1 copy
Naval Ship Research and Development
Computation and Mathematics Department
Bethesda, Maryland 20084

Captain Grace M. Hopper — 1 copy
NAICOM/MIS Planning Board
Office of the Chief of Naval Operations
Washington, D.C. 20350

Mr. Kin B. Thompson — 1 copy
Technical Director
Information Systems Division
OP-91T
Office of the Chief of Naval Operations
Washington, D.C. 20350

Advanced Research Project Agency — 1 copy
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Professor Omar Wing — 1 copy
Columbia University in the City of New York
Department of Electrical Engineering and
Computer Science
New York, New York 10027

Office of Naval Research — 1 copy
Assistant Chief for Technology
Code 200
Arlington, Virginia 22217

## ABSTRACT

## UNDERSTANDING GOAL-BASED STORIES

Robert Wilensky

Yale University

A reader

Reading requires reasoning. When a person reads, he often needs to infer connections between the sentences of a text, and must therefore be capable of reasoning about the situations to which the text refers. People can are able to reason about situations because they possess a vast store of knowledge, which They can use their knowledge to infer implicit parts of a situation from those aspects of the situation explicitly described by a text.

PAM (Plan Applier Mechanism) is a computer program that understands stories by reasoning about the situations they reference. PAM reads stories in English, and produces representations for the stories that include the inferences needed to connect each story's events. To demonstrate that it has understood a story, PAM answers questions about the story and expresses the story from several points of view.

PAM reasons about the motives of a story's characters. Many of the inferences needed for story understanding are concerned with finding explanations for events in the story. PAM has a great deal of knowledge about people's goals, which it and applies this knowledge to find explanations for the actions taken by a story's characters in terms of that character's goals and plans.

PAM is based on a theory of explanation that has two aspects. One part of this theory describes the types of goal-based story situations. Of particular interest are those situations involving a number of interrelated goals. The second aspect of the theory is a method of applying this knowledge to make inferences. This method combines a predictive component with a bottom-up inference mechanism to produce those inferences needed to understand a story without also producing many superfluous or erroneous inferences along the way.

## Preface

Understanding is an active process. This process is so natural to people that they are usually unaware of doing much when they read. However, when we try to program a computer to perform the same tasks that people find so trivial, the complexity of understanding becomes evident. People continually draw upon their experience and knowledge to understand language utterances and respond to them intelligently. For a computer to comprehend what it reads, it too must have knowledge about the world.

To give a computer the knowledge it needs to understand stories, we need to know what this knowledge looks like. Since stories are usually about people, a computer program needs to know about the human world, which is full of purposeful actors. For a program to understand stories about people, then, it must understand people's goals. It must have knowledge about the situations that intentional beings may find themselves in, and understand how they can be expected to act in these situations. By equipping a computer program with knowledge of human intentionality and by giving it a means to apply this knowledge, it is possible to build a machine that understands stories involving complex interactions between people.

## Acknowledgments

My advisor, Professor Roger Schank, sparked my interest in natural language processing, and constructed an environment in which creativity and thoroughness are both valued. His persistence, insights, and enthusiasm are largely responsible for my own.

Professor Robert Abelson served on my committee, and together with Professor Schank, produced many of the ideas that my own work is based upon. His intelligence and wit added greatly to my experience at the Yale A. I. Project.

Dr. Christopher Riesbeck taught me most of what I know about A. I. programming, and spent almost as much time showing me the errors of my ways as he did thinking up puns. A special thanks to him for his help and encouragement, and for translating much of my thesis into English.

Professors Drew McDermott, Gene Charniak, and Wendy Lehnert, Professor Edward Smith of Stanford, and Larry Birnbaum read preliminary drafts of this thesis, and provided much needed comments and corrections. I would also like to thank Professor Ned Irons for serving on my reading committee along with Professors Schank and Abelson.

Many other people at the Yale A. I. Project contributed to the work presented in this thesis. Janet Kolodner, Rick Granger, Rod McGuire, and Mike Lebowitz all helped my program grow and communicate with theirs. Walter Stutzman kept our machine alive and was particular helpful in the final preparation of this manuscript. Thanks to Jaime Carbonell, Richard Cullingford, Jerry DeJong, Anatole Gershman, James Meehan, and Mallory Selfridge, with whom I engaged in many interesting and beneficial discussions.

--------------------

The research described in this thesis was supported by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under contract N00014-75-C-1111.

TABLE OF CONTENTS

# CHAPTER 1

# WHY PAM?

Stories usually describe situations involving purposeful interactions between people. To understand stories, a reader needs a large body of knowledge about the kinds of situations that may occur, and must be able to use this knowledge to infer implicit aspects of a situation described by a text. Building a computer program that understands stories therefore requires an explicit description of this knowledge, and the construction of an inference mechanism with which to apply it.

This thesis is about understanding story situations involving goals. Most of the thesis is concerned with describing the categories of story situations that goals arise in, and with giving algorithms for detecting and processing the situations in each category. The most interesting categories of story situations involve multiple goals. Understanding a situation with a number of goals in it requires a reader to determine the interrelationships between those goals, and to infer the consequences of these relationships. Goal relationships form the basis for most of the situational categories presented here.

Knowledge of story situations and how they can be processed was used to build a story understanding program called PAM (Plan Applier Mechanism). PAM reads stories in English, and produces story representations that include the inferences needed to connect each story's events. To demonstrate that it has understood a story, PAM answers questions about the story and expresses the story from several points of view.

This chapter introduces the problems involved in understanding goal-based stories. Several previous theories of contextual language understanding are examined to see how they treat these problems, and the following conclusions are reached: The knowledge about story situations as well as the inferential capabilities needed for using it are not specified by these theories. A brief description of this knowledge is presented, and some of the problems involved in its application are discussed.

## 1.1 Introduction

For a program to understand a natural language text, it must be capable of reasoning about the situations that the text describes. For example, consider the following story:

(1) One day, John went through a red light and was pulled over by a cop. John had just gotten a summons for speeding the previous week, and was told that if he got another violation, his license would be taken away. Then John remembered that he had two tickets for the Giants' game on him. He told the cop that he would give them to him if he would forget the whole incident. The cop happened to be a terrific football fan. He took John's tickets and drove away.

Q1) Why did John offer the cop a couple of tickets?
A1) Because he was afraid he was going to lose his license if he got another summons.

To answer questions like Q1 about story (1), a reader must have understood the following aspects of the situation described in the story:

1. The cop was probably going to give John a summons for a traffic violation.

2. The authorities would then uncover John's poor driving record and take away his license.

3. John, not wanting to lose his license, tried to prevent the cop from issuing a ticket by making a deal with him.

4. The cop liked the deal, so he decided to accept it.

5. The cop didn't give John a summons.

The difficulty with understanding these aspects of story (1) is that they are not mentioned in the story text. A reader would have had to infer them from the facts explicitly stated in the story. Only by assuming that John didn't want to get another ticket and thus lose his licence (inference 3 above) could the reader have answered question Q1.

On the surface, these inferences do not appear particularly troublesome; story (1) is rather straightforward and is easily comprehended by most human readers. However, most existing natural language systems would be incapable of reading story (1) and answering question Q1. They would not understand story (1) because they would not be able to make the necessary inferences.

Consider what is involved in making the inference that John offered the cop his football tickets to prevent the loss of his license. First, the reader would have to infer that the cop was going to give John a

traffic ticket. This inference requires the knowledge that a policeman is supposed to ticket people who break traffic rules. Without this knowledge, story (1) would seem as reasonable as

(2) One day, John went through a red light and was pulled over by a milkman.

Unlike story (1), story (2) is peculiar because there is no obvious explanation for the milkman's pulling John over, and no obvious expectation for what he might do next. In fact, a reader might even hypothesize that the milkman was a cop in disguise in order to explain the sentence.

Next, the reader must infer that John didn't want to lose his license. For this inference, the reader needs to know the kinds of things people might want to happen, and the kinds of things they might want to prevent from happening. Losing something that has a useful function is something people would normally try to prevent. Since a license has the useful function of allowing its owner to drive a car, it can be inferred that John probably wanted to keep his license. Without this information about the usefulness of a license, the second sentence of story (1) would seem no different from

(3) John had just gotten a speeding ticket the previous week, and the police had told him that if he got another violation, they would take away all his garbage.

Having inferred that John wanted to hold on to his license, the reader must then infer that preventing the cop from issuing a summons would prevent the authorities from taking away his license. This inference is based on the principle that if one event can cause another event, then a way to prevent the second event is to prevent the first.

Now the reader must interpret John's statement to the cop as an attempt to prevent him from giving him a ticket. To interpret this sentence as an offer, a reader must know that one way to prevent someone from doing something is to persuade him not to do it. A person may be dissuaded from an action by offering him something desirable in exchange for his cooperation. The understander can infer that football tickets are desirable to a football fan, since football tickets are necessary for getting into a football game.

To see that all these inferences are in fact necessary, look what happens if we violate some of their premises:

(4) John told the cop he would give him some stubs from last week's football game if he would forget the whole incident.

(5) John told the cop he would give him the football tickets if he would forget the whole incident. The cop happened to despise football. He took John's tickets and left.

Story (4) cannot be interpreted as a method of persuasion because stubs, unlike tickets, are usually worthless. Story (5) is odd because the cop shouldn't have found the tickets desirable, yet he appeared to be persuaded by the offer.

The inferences needed to understand story (1) require knowledge about <u>why</u> people do things. Previous theories specify neither the knowledge required for such inferences, nor the means for applying this knowledge so that the correct inferences are made without generating many spurious or irrelevant inferences. Such theories could not be used to build a computer program to effectively make the inferences needed for understanding this story.

## 1.2 PAM

PAM (Plan Applier Mechanism) is a program that uses knowledge about intentions to make inferences. The program demonstrates its understanding by answering questions about the stories. In addition, since the program keeps track of each character's goals, the stories can also be paraphrased from different points of view.

Here are some examples of PAM input and output·

******************************************************************************

INPUT TEXT:

John wanted money.
He got a gun and walked into a liquor store.
He told the owner he wanted some money.
The owner gave John the money and John left.

INPUT: Why did John get a gun?
OUTPUT: Because John wanted to rob the liquor store.

INPUT: Why did John threaten the shopkeeper?
OUTPUT: Because John needed to get some money.

INPUT: Why did the shopkeeper give John the money?
OUTPUT: Because the shopkeeper didn't want to get hurt.

INPUT: Why did John leave?
OUTPUT: Because John didn't want to get caught.

OUTPUT:

STORY FROM JOHN'S POINT OF VIEW:

I needed to get some dough. So I got myself this gun, and I walked down to the liquor store. I told the shopkeeper that if he didn't let me have the money then I would shoot him. So he handed it over. Then I left.

OUTPUT:

STORY FROM THE OWNER'S POINT OF VIEW:

I was minding the store when a man entered. He threatened me with a gun and demanded all the cash receipts. Well, I didn't want to get hurt, so I gave him the money. Then he escaped.

************************************************************************

INPUT TEXT:

Willa was hungry.
She picked up the Michelin Guide
and got into her car.

INPUT: Why did Willa pick up the Michelin Guide?
OUTPUT: Because Willa wanted to know where a restaurant was.

INPUT: Why did Willa get into her car?
OUTPUT: Because Willa wanted to get to a restaurant.

INPUT: What were the consequences of Willa picking up the Michelin Guide?
OUTPUT: This enabled Willa to read the Michelin Guide.

OUTPUT:

THE STORY FROM WILLA'S POINT OF VIEW:

I wanted to get something to eat, but I didn't know where a restaurant was. So I picked up the Michelin Guide, and I got into my car.

*****************************************************************************

INPUT TEXT:

John and Mary were married.
Then one day, John was killed in a car accident.
Mary had to get a job.

INPUT: Why did Mary need employment?
OUTPUT: John died and so she needed a source of money.

*****************************************************************************

INPUT TEXT:

John wanted to watch the football game,
but he had a paper due the next day.
John watched the football game.
John failed Civics.

INPUT: Why did John fail a course in Civics?
OUTPUT: He failed to hand in an assignment.

*****************************************************************************

To build PAM, I first had to develop a theory of inference. The arguments in this thesis supporting this theory are presented from the point of view of a hypothetical story reader that represents my intuitions about the nature of human natural language processing. I refer to this hypothetical entity as "the understander" or "the reader". Wherever possible, arguments are aimed at describing characteristics of the reader that must hold whether the reader is human or mechanical. I consider PAM to be a implementation of this reader.

## 1.3 Related Work

Because the problem of inference is so crucial for understanding even the simplest of stories, it has begun to attract a great deal of attention from Artificial Intelligence natural language researchers (for example, see Charniak (1972), Schank (1974), Rieger (1975), Wilks (1975), Joshi (1976), Cullingford (1978)). Before proposing answers to the above questions, it is useful to make a critical examination of some of the existing theories of inference.

### 1.3.1 Rieger's model

One of the earliest working inference systems was that of Rieger (1975). Rieger, whose program was the MEMORY component of the MARGIE system (Schank, et al. 1973), proposed sixteen classes of inference needed for language understanding. Some examples of his inference classes are:

1. Resultative Inference:

   Input: John gave Mary a car.
   Inference: Mary has the car.

2. Motivational Inference:

   Input: John hit Mary.
   Inference: John probably wanted Mary to be hurt.

3. Functional Inference:

   Input: John wants a book.
   Inference: John probably wants to read the book.

4. Feature Inference:

   Input: Andy's diapers are wet.
   Inference: Andy is probably a baby.

Schank (1973) suggested that chains of causal links are useful for describing the connection between sentences of a text. Since some of Rieger's inferences generate causal connections, it should be possible to relate two sentences in a text by generating causal inference upon causal inference until a connection between the two sentences is established. The problem lies in the generation of these inferences.

Rieger conjectured that a natural language understander actually made <u>all</u> the possible inferences from a given input. An inference procedure associated with each class of inference examined the input, and if it met that procedure's criteria, the procedure made its inference. Inferences were then made from these inferences, and so on. Associated with each inference was a strength, which grew weaker with each generation of inferences. The process terminated when the strength of the inferences being generated fell beneath some arbitrary cut-off value.

Rieger estimated that 500-1000 inferences would be generated during the comprehension of a simple utterance. Admitting the computational complexity of this process, Rieger placed his faith in the development of powerful parallel hardware and smart "forgetting" mechanisms to prevent the demands of the inference maker from overloading the system.

There is another problem with Rieger's inference model besides its awkwardness and computational cost. If we look back at the inferences needed to understand story (1) in the beginning of this chapter, the need for an inferential capability beyond Rieger's theory becomes apparent. For example, the inference that the policeman was going to give John a summons is based on knowledge about the police and how they are expected to act in particular situations. Rieger's work does not suggest how to make this inference, or how to cope with the knowledge that such an inference requires. The problem here is that Rieger's inference machinery is oriented toward making immediate, "low-level" inferences about an event, i.e., those inferences that are conceptually very close to the event itself. It does not address the problem of how to bridge the larger inferential gaps that appear in stories. These inferences appear to require a great deal of knowledge about situations or the way people might act in them.

### 1.3.2 Cullingford's model

A number of attempts have been made to address the problem of controlling the inferential explosion of Rieger's system, as well as to provide the additional inference abilities needed for story understanding. For example, the SAM program (Cullingford, 1978) uses knowledge of stereotyped situations to prune inference making. Cullingford's program stores its knowledge in the form of <u>scripts</u> (Schank and Abelson, 1977), which are mundane sequences of events. Once SAM realizes that a particular script is being referred to, it uses the script to interpret further inputs and make inferences.

As an example, the program might contain in its memory a script describing what happens when one goes to a restaurant. Upon hearing a story about someone going to a restaurant, SAM would use the description of the events found in the restaurant script to fill in pieces of the story not mentioned in the actual text. Suppose SAM were told the following story:

(6) John went to a restaurant. He ordered lobster. When the check came, John paid it and left a large tip. John left the restaurant.

SAM tries to match each of the events in the story against the events encoded in the restaurant script. In the process, SAM fills in missing events from the events stored in the script. In story (6), SAM infers that John ate the lobster because this event appears in the script between events of ordering and getting the check.

Unlike Rieger's program, SAM does not have to generate many inferences in order to make this particular inference. SAM is able to exert such control over the inference process because it has a great deal of specific knowledge about the situation being described in the story. Once SAM has a script available, it does not look beyond the content of that script to make inferences.

### 1.3.3 Charniak's model

SAM is an example of a trend in AI toward "larger" chunks of knowledge than those found in Rieger's program. As such, SAM is a good example of what Minsky has called a frame system (Minsky, 1974). A frame is a hierarchical data-structure for representing stereotyped knowledge, be it about situations or physical objects. At the terminals of a frame are slots that can be filled by certain kinds of entities (usually other frames). These terminals are normally filled with "default" values.

While the script concept underlying SAM was developed independently from the notion of frames, Charniak has written a language understanding program more explicitly based on the frame system idea (Charniak. 1978). Ms. Malaprop, as his program is called, uses "framed" knowledge to understand and answer questions about simple texts involving painting (as in painting a chair). Like SAM, Ms. Malaprop uses its frames to bridge the gaps between story events without having to resort to a combinatorial explosion of inferences.

From the viewpoint of inference making, Ms. Malaprop bears a great resemblance to SAM. Inputs are matched against a frame describing the situation underlying the story, and the frame is used similarly to a script to fill in missing pieces. Like SAM, Charniak's program restricts its inference making to those inferences readily available from a fixed structure.

### 1.3.4 Problems

It is also not surprising, then, SAM and Ms. Malaprop share a number of serious deficiencies with their inference mechanisms. In fact, their creators were first to point out these problems. Probably the most compelling problem is that of generality. If all stories conformed to the structure of a script (or to one of Charniak's frames), then a SAM-like program might well be the ultimate understanding system. However, suppose the sequence of events in the story were somewhat unusual, but still understandable. Since all of SAM's knowledge is based on stereotyped situations, the program would be at a loss to find the connections between the events of the story.

For example, consider the following simple story, adapted from Schank and Abelson (1977):

(7) Willa was hungry. She picked up the Michelin Guide.
    Willa got into her car.

A reader of story (7) must infer that Willa picked up the Michelin Guide to read it in order to find out where a restaurant was. Then the reader could infer that Willa got into her car in order to drive to a restaurant she had found in the guide.

Story (7) presents a problem for a program like SAM. For SAM to understand (7), it would need a script that essentially corresponded to the story. That is, the script would contain the following events:

"Person be hungry"

"Person pick up guide book"

"Person read guide book"

"Person select a restaurant"

"Person get into vehicle".

"Person drive to restaurant"

The problem with this solution is that the program would need a script for every conceivable situation it might hear about. The number of scripts would not just be large; it would be unlimited. People are constantly reading about situations that they are unfamiliar with. In fact, there would be little motivation for a person to read at all if every story conformed completely to a stereotyped situation. Even if an understander happened to have the above script sitting around when story (7) came along, eventually some variant of the story would be encountered for which no script was available.

Not only is the notion that all stories conform to scripts highly implausible, but it mitigates the advantage that the script approach has over a Rieger-type memory. If SAM had scripts for all the possible things a person might do, then the generation of all possible inferences in Rieger's program would be replaced by the task of searching through the multitude of scripts contained in the data-base.

This criticism is applicable to Ms. Malaprop as well. For example, consider the following story that Ms. Malaprop had trouble understanding:

(8) Jack was going to paint. He washed the brush.

Presumably, Jack washed the brush to make sure it was clean. Malaprop was not able to find this interpretation because her painting frame does not have in it a particular event describing brush cleaning at the beginning of the frame. In spite of the fact that the reason for cleaning the brush is apparent to most readers, the program could not infer it because it had not been explicitly stated within the painting frame. As is the case for SAM, this particular problem could be patched by making an addition to the painting frame. But the frame loses its inferential advantage if it is expanded to encompass all possible stories.

It is evident that the script idea has some application to story (1), but that it does not solve the entire problem. For example, a reader might have a script for getting a summons, and use this knowledge to explain the cop's pulling John over. But it is unlikely that this

script would cover all story episodes that could possibly follow this event.

### 1.3.5 Other Work on Inference

Wilks (1975) showed how an inference mechanism could be naturally introduced into his theory of preferential semantics. The integration of an inference mechanism into his system was so natural, in fact, that it is difficult to come to a separate evaluation of it. Basically, Wilks' system understands text by trying to fit natural language inputs into formal structures, called templates and paraplates, that are defined in terms of a number of semantic primitives. In this system, inference is just another technique for finding the appropriate structures for a given language utterance.

Several problems arise in comparing Wilks' work with those mentioned previously. First, his work has been oriented toward the view that a system should understand a text only as much as is needed for the particular natural language task at hand. In his case, this task has been machine translation. It is by no means clear that the level of understanding required by this task will be sufficient for other natural language tasks, such as question answering. Second, his concern has been to show how inference rules can be integrated into his system, rather than the nature of the knowledge used to make inferences. The focus of my own work, and that of Rieger, Schank, Cullingford and Charniak, has been more in the direction of showing how the kinds of knowledge available to the system determines the nature of the inference mechanism and the kinds of inferences produced.

Thus it is difficult to talk about the nature of Wilks' theory of inference since he is not committed to a particular class of inferences. For example, Wilks has recently introduced into his inference mechanism the notion of a <u>pseudo-text</u> (Wilks, 1977), which seems similar in flavor to a script or a frame, and very different from the knowledge sources previously available to his understander. Since his system is accommodating enough to support a variety of knowledge structures, it is hard to compare it to systems more committed to particular kinds of knowledge.

A similar problem arises in discussing the work of Joshi and Rosenschein (1976) and Joshi and Weischedel (1977). Their research is largely concerned with uncovering the formal properties and structure of an inference system, rather than with the properties and structure of knowledge that the system uses. Furthermore, to my knowledge, they have not yet applied their ideas to story comprehension, so a direct comparision of ideas is not possible.

In sum, story understanding systems are faced with the problem of controlling the proliferation of inferences, i.e., of making the right inference at the right time without making a multitude of erroneous or irrelevant inferences. This problem can be partially addressed by providing the understander with knowledge about stereotypical situations. However, this solution does not address the question of how

to build a flexible language understander that could make correct inferences in unfamiliar contexts.

## 1.4 Goals

What can be done to remedy this situation? What needs to be done in order to build a general understanding system? To answer these questions, it is useful to examine the source of the limitations of a story understander like SAM.

The primary reason that SAM is limited in its generality is that its inference mechanism is based on knowledge of common sequences of actions. However, a great number of the inferences needed to understand simple situations are not based on such knowledge. For example, consider the following stories:

(9) John loved Mary. One day, John saw a truck coming down the street toward Mary. John ran up behind Mary and gave her a shove.

(10) John hated Mary. One day, John saw a truck coming down the street toward Mary. John ran up behind Mary and gave her a shove.

Suppose after reading each story, an understander were asked what John did. In story (9), the most reasonable answer seems to be that he tried to push her out of the way of the truck. On the other hand, in (10). the reader would probably assume John tried to push Mary in front of the truck.

These inferences are based not so much on the reader's knowledge of events (which are pretty much the same in both stories), but on the reader's knowledge of people and how they may relate to one another. To understand stories (9) and (10), the following form of knowledge is required:

A person may want to harm someone he greatly dislikes.

If a person loves someone, then he will want to help that person if that person is endangered.

Being hit by a large moving object can harm a person.

The first two rules are about <u>why</u> people do things. SAM does not understand the <u>reason</u> behind a person's behavior. For example. SAM's restaurant script

(11) John went to a restaurant. The waiter came over and said they were out of menus. John asked the waiter what he recommended.

To understand (11), a reader must recognize that the point of a menu is to inform the customer what dishes the restaurant prepares, and that this information can also be conveyed in other ways.

SAM could not have understood (11) because SAM does not know why a waiter brings the menu. From SAM's viewpoint, reading the menu in a restaurant is a ritual, just as if prayer were required before ordering.

SAM does not have knowledge about intentions because this knowledge is not needed to understand stereotyped situations. However, once a story violates a stereotype, this information becomes crucial. In order to understand story (11), a reader must know the intentions behind the actions taken by the characters. Underlying each action in a script is a reason why a character needs to perform that action. For example, the waiter brings the menu in order to tell the customer what the restaurant has. Once a reader realizes that this is the reason for a waiter's action, he can understand how a non-standard event might also be justified by the same reason. In story (11), a non-standard event was John asking the waiter to tell him about the restaurant's dishes. Asking someone to tell you something is a way of finding out something. Since this accomplishes the same purpose as reading a menu, a reader could understand why John asked the waiter a question in place of reading a menu, even though there is no such an event in the restaurant script.

To understand intentions, a reader must be able to determine the state of affairs a character desires to bring about by performing an action. This desired state of affairs is called a goal. In story (11), the understander needs to infer John's goal of knowing what the restaurant prepares. A way of accomplishing a goal is called a plan. In (11), John uses the plan of asking someone to tell him something and then waiting for a response.

The story reader's main job is to understand the goals of each character in the story. This includes inferring characters' goals, determining why a goal is present, inferring the plans each character is using to achieve his goals, and noticing how characters' goals relate to one another.

Knowledge of how goals may be achieved is not necessarily tied to particular situations. For example, in story (11), John had a goal of finding out some information. He attempted to accomplish this goal by asking someone who was likely to know. In effect, John was using a plan of great generality. An understander who knew that asking is a plan, and knew the goals for which asking may be used, could understand (11) as containing an instance of this general plan. If we were then to ask such an understander why John asked the waiter what he recommended, the understander could answer with the goal underlying this event, namely

(12) John wanted to know what the restaurant had to offer.

Thus a reader can understand a story in terms of the plans and goals of the story's characters <u>even if the events constituting the plans do not conform to stereotyped sequences</u>.

We are now in a better position to analyze the inferences required by story (1):

(1) One day, John went through a red light and was pulled over by a cop. John had just gotten a summons for speeding the previous week, and was told that if he got another violation, his license would be taken away. Then John remembered that he had two tickets for the Giants' game on him. He told the cop that he would give them to him if he would forget the whole incident. The cop happened to be a terrific football fan. He took John's tickets and drove away.

In this story, John had the goal of preserving his right to drive. Instrumental to this goal was the goal of getting the cop not to give him a summons. John decided to use the general plan of bargaining with the cop to dissuade him from doing his job. This plan requires that its user has something that the other person might want. A person might want tickets for a football game if he is a football fan because he might have the goal of watching the football game, and having football tickets is instrumental to this goal. Thus we can understand John's offer to the cop as an instance of the bargaining plan for John's goal of not getting a summons.

The need to follow the goals of a story's characters poses a number of questions for a builder of a natural language understanding system. What kinds of goals are there? How can we tell when a goal is present? When one has been fulfilled? Abandoned? Set aside? What kinds of plans are there for these goals? How do we determine which plan a character is using? What goal is behind this plan? How do the goals of the various characters in a story relate to one another? In general. what kind of knowledge about intentionality is needed for a natural language understanding system, and how can this knowledge be applied to the understanding task?

## 1.5 Applying knowledge about goals

Theoretical answers to some of the above questions are suggested in Schank and Abelson (1977). They provide a classification of goals and plans that can be used to describe situations such as the ones that concern us here. I made use of a great deal of their theory in my work, and will discuss and criticize this theory in detail in Chapter 3.

In this section I want to address some of the problems that arise in the application of intentional knowledge to the task of story understanding. The point of this section is to demonstrate some of the characteristics that a story understanding mechanism must embody to

apply knowledge of goals and plans successfully.

In other story understanding systems, like SAM or Ms. Malaprop, most of the processing goes on in an extremely top-down mode. That is, the program has a data structure that it wants to fill out, and spends its time determining where each input could be placed in that structure. For example, SAM has a script it is trying to instantiate (i.e., match to a story), and spends its time matching incoming sentences to events encoded in the script. The top-down component of a system often functions as a predictive mechanism. SAM predicts that the next event in a story will be one of the next events that follow the current event in the script.

A predictive mechanism is in contrast to bottom-up (also called data driven) processing. Bottom-up programs work by examining an input and then deciding what to do with it, rather than trying to fit the input into a preconceived template. For example, Rieger's MEMORY program was bottom-up because the inferences it made were strictly a function of the input it received. Bottom-up programs can be said to function as a recognition mechanism. Rieger's program recognized an input as belonging to a particular inference class, and then made the inferences characteristic of that class.

Unfortunately, neither of these notions is sufficient for a program that is to understand non-stereotypical stories. While top-down processing greatly constrains the making of inferences in understanding stories, a purely top-down system has trouble with novel situations. To understand novel stories, a mechanism must first determine what data structure it should be trying to fill in.

The bottom-up approach is reasonable at the beginning of stories such as

(13) John asked Bill where a restaurant was.

A reader of (13) readily infers that John probably wanted to know where a restaurant was so he could go there and get something to eat. The inference follows from recognition of the ASK plan, which is being used here to find out information. Knowing the location of something is instrumental to going there. In this case, the desired location is a restaurant, and being at a restaurant is instrumental to doing the restaurant script to satisfy one's hunger.

So without a context a story understander can infer the goals and plans of a character in the story. However, a purely bottom-up inference mechanism does not give a good account of how to connect story events. Suppose (13) were followed by

(14) Bill said there was one about a mile away.

Sentence (14) hardly seems as if it could be processed in a strictly bottom-up fashion, as was (13). To do so would require an inference system that always tried to interpret someone telling someone something as an answer to a question, even if the context suggested the sentence was a response to a threat or some other plan. Only by hypothesizing

this interpretation every time would a bottom-up system be able to recognize the connection when it does arise. This is exactly the situation which programs like SAM were designed to avoid. Moreover, it seems entirely reasonable that once a question is encountered in story understanding, a predictive (i.e., top-down) process is invoked that looks to see if the next input is a response to a question, and decides what to do with such an input if it is.

So far it seems an understander has to be bottom-up until it has some idea of what the story is about, i. e., it has accessed some knowledge relevant to the story. Unfortunately, this is not the case. For suppose (14) were followed by one of the following:

(15) John got on the bus.
(16) John asked Bill where he could catch the bus.
(17) John asked Bill for a lift.

We need a top-down prediction regarding travel, but there are far too many possible ways of going, not to mention actions related to going, such as asking for help. Therefore, we really need to infer bottom-up from specific actions such as (15), (16) and (17) until we recognize that novelty is involved.

To sum up so far: Some kind of recognition mechanism seems to be needed for understanding the plans and goals underlying the events of a story. This mechanism is needed because an understander can't always have an exact idea of what might happen next. On the other hand, the reader is not totally unbiased about these events either, for he often has some predictive notion about the kinds of actions a character might take. The reader can use this knowledge to help interpret the connection between events (e.g., to determine that (17) is related to John's goal of being at a restaurant). The question is, how can an understander predict something just a little? How can it have a prediction specific enough to interpret (17) as being related to John's goal, but vague enough to allow for the variability so crucial for understanding non-stereotypical situations?

One solution to this problem requires combining a bottom-up inference mechanism with a predictive mechanism so as to make predictions about bottom-up inferences. Using this idea, prediction can be made about the goals or plans that will be inferred later on in the story. When subsequent events of the story are read, a bottom-up procedure can make inferences about the intention that underlies the event. Then the predictive mechanism can examine these inferences to determine if a prediction has been confirmed.

Thus predictions can be used to constrain the bottom-up inference process without eliminating it entirely. An algorithm that synthesizes these two processes is presented in the next chapter.

## 1.6 What this thesis is about

In the last two sections, I described some of the reasons that a reader needs knowledge about goals, and sketched some of the problems that arise in applying this knowledge. However, goals occur in many different types of situations. To understand goal-based stories, a reader must be able to determine which knowledge is applicable to a given story, and have a means for applying it.

This thesis is concerned with understanding story situations involving goals. Several categories of story situations are presented, organized by the knowledge and inference rules needed for identifying and processing the situations in each class.

Most of these categories describe story situations that have multiple goals. Each character in a story will often have more than one goal, and most stories involve more than one character. The most crucial part of understanding a situation is often understanding the relationships between goals. Goal relationships form the basis for most of my situational categories.

For example, relationships between goals are important in stories involving goals that recur predictably. Consider the following story:

(18) John got tired of walking to work every day. He decided to buy a car.

(19) John got tired of walking to work every day. He decided to ask Bill if he could borrow his car.

To understand why John bought a car in story (18), it is necessary to realize that John will continue to have the goal of being at work every day. Moreover, owning the car makes it easy for John to use the car whenever he wants to. Story (19) is strange because it does not seem to be a solution to John's problem.

PROBLEM: What does an understander need to know about planning for frequently recurring goals? This is discussed in Chapters 4 and 5.

Relationships among goals are important when one character has several goals at the same time. For example, consider

(20) John wanted to watch the football game, but he had a paper due the next day. John decided to watch the game anyway. John failed civics.

What is the relationship between watching a football game and failing a course? Understanding story (20) requires knowledge of the <u>goal conflicts</u> that may surface when a character in a story has multiple goals. In story (20) the conflict existed because there wasn't enough time for John to do all the things he wanted.

PROBLEM: What do we need to know about planning in situations where a character has conflicting goals? Goal conflicts are discussed in Chapters 6 and 7.

A situation similar to goal conflict can occur among the goals of several characters. For example

(21) The Republican president thought that McLiberal would be a weaker candidate to run against than Mushie. He decided to help McLiberal win the Democratic nomination.

To understand the President's actions, a reader would have to realize that the goal of the Democratic nominee competes with the goal of the Republican candidate. Then, since a weak opponent is easier to defeat than a strong one, the President's action can be interpreted.

PROBLEM: What kinds of plans do characters use when their goals compete with one another? How can a reader recognize that two goals are in competition? Understanding competition is the subject of Chapters 8 and 9.

Sometimes several characters may work together toward a common goal. For example, consider the following story:

(22) John and Bill were partners playing golf. Bill hit a shot into the rough. John sneakily moved the ball into a better position.

To understand why John helped Bill, a reader would have to realize that John and Bill have a common goal because they are both members of the same team. Thus John's action can be explained in terms of a plan to help someone else whose goal concurs with his.

PROBLEM: How can a reader recognize that two goals are in concord? What plans can characters with concordant goals use to aid one another? Understanding goal concord is described in Chapter 10.

## 1.7 A PAM Example

PAM uses knowledge about goals and plans to understand novel stories. PAM has knowledge about the kinds of goals people have, how these goals come about, and how they may be fulfilled. PAM also has knowledge of the more complicated goal relationships mentioned in the previous section. PAM applies this knowledge to the story understanding task to make the inferences needed to find connections among sentences in a story.

To build PAM, I had to develop the theoretical apparatus needed to handle the complicated story situations discussed in the previous section. I also had to address the problem of building a system that combined the flexibility of a bottom-up inference mechanism with the frugality and predictive power of a goal-directed mechanism. PAM accomplishes these objectives by use of the following:

1. Small, general chunks of knowledge about plans, goals, goal relationships, etc.

2. A technique for accessing the knowledge relevant to each sentence of a story.

3. A mechanism that controls inferences made from each sentence depending upon the previous sentences in the story, and uses these inferences to establish the connections implicit among the story's sentences.

By using top-down prediction to control a bottom-up inference device, PAM can make the inferences needed to understand a piece of text even if the connections among the text's sentences are not particularly stereotyped. At the same time, PAM can make these inferences without making large numbers of irrelevant inferences.

The following is an abridged example of a PAM story understanding session. PAM leaves a trace containing the rules it uses to understand a story, and the plans and goals it infers to connect story events. The algorithm PAM uses is outlined in Chapter 2, and particular plans and goals known to PAM are discussed in Chapter 3. Chapters 13 and 14 describe the details of PAM's implementation. Thus the following trace refers to many items that are explained in subsequent chapters. The trace is include here primarily to give the reader some idea of the kinds of inferences PAM makes, and when it makes them.

```
[PHOTO: Recording initiated Tue 8-Aug-78 12:53PM]

@RUN PAM

*(UNDERSTAND CD1)

THE STORY IS

 JOHN WANTED MONEY.
 HE GOT A GUN AND WALKED INTO A LIQUOR STORE.
 HE TOLD THE OWNER HE WANTED SOME MONEY.
 THE OWNER GAVE JOHN THE MONEY AND JOHN LEFT.
```

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(JOHN WANTED MONEY) | PAM processes the story a sentence at a time. |

```
CONCEPTUALIZATION IS:
((CON
  ((ACTOR MONEY0 IS
    (*POSS* PART HUM0))
     TIME (FORM41))
   IS (*GOAL* PART HUM0))
 TIME (FORM42))
```

This is the input in Conceptual Dependency, the form actually manipulated by PAM. HUM0 is a token (representation for an individual) created by PAM for the character John. The information stored under tokens is shown in Chapter 14. The time forms (FORM41 and FORM42) contain time information in an internal format. For example, FORM42 states that John had this goal some time in the past. FORM41 states that his having money has not occurred at the time of this event.

```
FOUND PLAUSIBLE EXPLANATION

 GOAL: (*DCONT* PLANNER HUM0
OBJECT MONEY0 OWNER (NIL)
RECIPIENT HUM0)
 SOURCE:
(*SUBGOAL* PLAN (NIL))
(*THEME* VAL (*FONDNESS*
 OBJECT MONEY0))
(*GOAL-SUBSUMPTION* GOAL (NIL))
```

PAM now looks for an explanation for the input. This explanation denotes John's goal of getting some money. PAM also infers that John had this goal because it was instrumental to another goal, or because he likes money, or because he might need it in the future. The notion that something might come in handy for future recurring goals is called goal subsumption, and is treated in Chapters 4 and 5.

```
*** ADDING TO STORY REPRESENTATION:

 GOAL: (*DCONT* PLANNER HUM0 OBJECT
 MONEY0 OWNER (NIL) RECIPIENT HUM0)
```

Having found an explanation for the sentence, PAM adds the explanation to its representation of the story.

```
NEXT INPUT IS:
 (HE GOT A GUN)
```

PAM processes the next sentence.

```
CONCEPTUALIZATION IS:
((ACTOR HUM2 <=> (*ATRANS*)
 OBJECT PHYS3 TO HUM2)
 TIME (FORM47))
```

HUM2 denotes the pronoun "he". As PAM processes this input, it will determine that HUM2 refers to HUM1, which denotes the character John.

```
TESTING EXPLANATION

EXPLANATION IS GOAL:
(*DCONT* PLANNER HUM2 OBJECT PHYS3
 PHYS3 OWNER (NIL) RECIPIENT HUM2)
PLAN:
(*UNSPEC* PLANNER HUM2)
```

PAM finds a possible explanation for the input, namely that John had the goal of having a gun, and fulfilled this goal via some unknown plan. Now PAM tries to explain why someone might have this goal.

```
ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION

EXPLANATION IS PREDICT PLANS:
 *PB-THREATEN* *PB-OVERPOWER*

                 *
```

Functions attached under GUN cause PAM to predict that John will try to threaten or overpower someone. Since these are ways to get something from someone, PAM PAM infers that John will use one of these plans for his goal of getting some money.

```
*** ADDING TO STORY REPRESENTATION:

INFERRED GOAL: (*DCONT* PLANNER
HUMO OBJECT PHYS3 OWNER (NIL)
RECIPIENT HUMO)

 OUTCOME OF GOAL: (*DCONT* PLANNER
HUMO OBJECT PHYS3 OWNER (NIL)
RECIPIENT HUMO) IS (*SUCCEED*)

GOAL IS AN INSTANCE OF  GOAL:
(*DCONT* PLANNER HUMO OBJECT TOK1
OWNER (NIL) RECIPIENT HUMO)
```

PAM adds the inferred goal (John's wanting to have a gun) to the story representation, along with a few related inferences.

```
NEXT INPUT IS:
 (AND WALKED INTO A LIQUOR STORE)

CONCEPTUALIZATION IS:
((ACTOR HUM3 <=> (*PTRANS*)
  OBJECT HUM3
  TO (*INSIDE* PART ORGO)
  INST ((ACTOR HUM3 <=> (*MOVE*)
         OBJECT PARTO)
        TIME (FORM63)))
 TIME (FORM64))

TESTING EXPLANATION

EXPLANATION IS GOAL:
(*DPROX* PLANNER HUM3 OBJECT HUM3
  LOCATION (*INSIDE* PART ORGO))
PLAN:
(*PB-WALK* PLANNER HUM3
LOCATION (*INSIDE* PART ORGO))
```

PAM now determines that John plans to rob the liquor store.

```
*** ADDING TO STORY REPRESENTATION:

 INFERRED GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*INSIDE*
LOCATION (*INSIDE* PART ORGO))
```

PAM adds the inferences that John wanted to be inside the store,

```
 OUTCOME OF GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*INSIDE*
PART ORGO))  IS (*SUCCEED*)
```

that he fulfilled this goal

```
 PLAN: (*PB-WALK* PLANNER HUMO
LOCATION (*INSIDE* PART ORGO))
```

by walking into the store

```
GOAL IS AN INSTANCE OF  GOAL:
(*DPROX* PLANNER HUMO OBJECT HUMO
LOCATION (*PROX* PART MONEYO))
```

and that the goal was an instance of a more general goal of being near some money.

```
NEXT INPUT IS:
 (HE TOLD THE OWNER HE WANTED SOME
  MONEY)

CONCEPTUALIZATION IS:
((ACTOR HUM4 <=> (*MTRANS*)
 MOBJECT
 ((CON
  ((ACTOR MONEY1
    IS (*POSS* PART HUM4))
   TIME (FORM76))
   IS (*GOAL* PART HUM4))
  TIME (FORM77))
 TO (*CP* PART HUM1))
 TIME (FORM78))
```

After learning that John got a gun, PAM predicted that John would use the gun to threaten or to overpower somebody. These predictions cause PAM to test each input to see if it confirms one of them. The prediction that John will threaten someone with his gun is satisfied by what might otherwise be interpreted as a polite request. Thus this prediction causes PAM to interpret this input as a threat.

```
INFERRED PLAN:
 (*PB-THREATEN* PLANNER HUMO
  PERSUADEE HUM1
  THREAT ((ACTOR HUMO <=> (*PROPEL*)
           OBJECT HUM1 INST PHYS3)))

INFERRED  GOAL: (*PCONT* PLANNER
HUM1 OWNER ORGO OBJECT MONEY1)

INFERRED  GOAL: (*PHEALTH*
PLANNER HUM1 POSSESSOR HUM1)

GOALS ARE IN CONFLICT
```

PAM infers that the owner now has some conflicting goals: preserving his money and preserving his health. Goal conflicts are discussed in Chapters 6 and 7.

```
INFERRED GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*PROX*
PART TOK2))

INFERRED GOAL: (*PSOCCONT*
PLANNER HUMO OBJECT HUMO)
```

PAM also infers that John has the goal of preserving his freedom now that he has broken the law.

```
NEXT INPUT IS:
 (THE OWNER GAVE JOHN THE MONEY)

CONCEPTUALIZATION IS:
((ACTOR HUM5 <=> (*ATRANS*) OBJECT
  MONEY2 TO HUM6) TIME (FORM94))

INFERRED PLAN: (*PB-COMPLY*
PLANNER HUM1)

OUTCOME OF GOAL: (*DCONT* PLANNER
HUMO OBJECT MONEY1 OWNER ORGO
RECIPIENT HUMO) IS (*SUCCEED*)

OUTCOME OF GOAL: (*PCONT*
PLANNER HUM1 OWNER ORGO OBJECT
MONEY1) IS (*ABANDON*)
```

PAM now infers that the owner has complied with John and abandoned his goal of preserving his money, and that John has fulfilled his goal of having some money. Giving John the money is an instance of a plan for protecting oneself against someone who has a competing goal. The notion of competing goals and planning against the goals of another character are discussed in Chapters 8 and 9.

```
NEXT INPUT IS:
 (AND JOHN LEFT)

CONCEPTUALIZATION IS:
((ACTOR HUM7 <=> (*PTRANS*)
  OBJECT HUM7 TO (NIL)
  FROM (*INSIDE* PART (NIL))
  INST (NIL))
 TIME (FORM100))

INFERRED PLAN: (*PB-WALK*
PLANNER HUMO LOCATION (NIL))

OUTCOME OF GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (NIL)
IS (*SUCCEED*)

OUTCOME OF GOAL: (*PHEALTH* PLANNER
HUM1 POSSESSOR HUM1) IS (*SUCCEED*)
```

PAM had inferred that John had the goal of preserving his freedom, and therefore now infers that he is leaving to avoid getting caught. Also because John's leaving terminates his threat toward the owner, PAM infers that the owner's self-preservation goal has been fulfilled.

```
FINISHED UNDERSTANDING PHASE
```

```
************************************************************************
*                                                                      *
*  PAM has now produced a story representation that encodes the orig-  *
* inal input together with the inferences it made to understand the    *
* story. To demonstrate the level of understanding that this repre-    *
* sentation embodies, PAM can answer questions about the story.  PAM   *
* finds the answer to a question by examining the information encoded  *
* in the story representation.  For example, to answer "Why did John   *
* get a gun?", PAM looks for in the story representation for the event *
* of John getting a gun, and checks to see if a reason for event is    *
* recorded. PAM finds that John got the gun as part of a plan to rob a *
* liquor store. This answer is then expressed in English.              *
        *
*                                                                      *
************************************************************************
```

QUESTION: Q1

Why did John get a gun?
Because John wanted to rob the liquor store.

QUESTION: Q2

Why did John threaten the shopkeeper?
Because John needed to get some money.

QUESTION: Q3

Why did the shopkeeper give John the money?
Because the shopkeeper didn't want to get hurt.

QUESTION: Q4

Why did John leave?
Because John didn't want to get caught.

QUESTION: Q5

What were the consequences of John getting a gun?
John had a weapon which enabled him to rob the liquor store.

QUESTION: Q6

What were the consequences of John threatening the shopkeeper?
The shopkeeper didn't want to get hurt so he gave John the money.

QUESTION: Q7

What were the consequences of the shopkeeper giving John the money?
John had some money.

QUESTION: Q8

How did John go to the liquor store?
John walked to the liquor store.

QUESTION: Q9

How did John threaten the shopkeeper?
John got a gun and he walked to the liquor store.

QUESTION: Q10

How did John leave?
John walked.

```
************************************************************************
*                                                                      *
*  The story can also be expressed in English from several points of   *
* view:                                                                *
*                                                                      *
************************************************************************
```

WHO SHOULD TELL THE STORY? *JOHN

I needed to get some dough. So I got myself this gun, and I walked down to the liquor store. I told the shopkeeper that if he didn't let me have the money then I would shoot him. So he handed it over. Then I left.

WHO SHOULD TELL THE STORY? *THE OWNER

I was minding the store when a man entered. He threatened me with a gun and demanded all the cash receipts. Well, I didn't want to get hurt, so I gave him the money. Then he escaped.

[PHOTO: Recording terminated Tue 8-Aug-78 12:55PM]

PAM understands the liquor store story by applying its general knowledge of human intentionality, not by having particular knowledge about liquor stores or robberies. Since PAM's knowledge is not bound to

specific situations, PAM can use its knowledge to understand people's actions in entirely different domains.

## 1.8 Summary

A story understander is required to make inferences because stories often leave as much implicit as they state directly. A number of mechanisms exist that can perform some kind of inference. However, these mechanisms cannot make all the inferences needed for story understanding. In particular, the problem of how to understand a non-stereotypical story poses a great difficulty.

To build a general story understander, it is necessary to build a program that has a great deal of knowledge about human intentionality. This program needs to know about the kinds of goals people have, and how these goals might be fulfilled. It also need to know about complicated goal situations, such as goal conflicts and goal competition.

Furthermore, because of the nature of the program's knowledge, the process by which such a program should apply this knowledge must be different from that used in other inference mechanisms. In particular, the mechanism requires the predictive power of top-down processing while maintaining the responsiveness of a data-driven system.

# CHAPTER 2

## Overview of the Understanding Process

## or, Explanation-Driven Understanding

### 2.1 Introduction

PAM is a program designed to understand novel situations. In building PAM, I found that I had to build a program that knew a great deal about human intentionality. That is, PAM requires a large quantity of knowledge about people's goals.

Most of this thesis is concerned with describing the knowledge about goals I found I needed, and the algorithms I had to construct to apply this knowledge to the task of story understanding. Before I ask the reader to become immersed in the detailed structure of this knowledge, it is important to explain why all this is relevant to building a story understander. What is the process of story understanding like? Why should it require all this knowledge about goals?

This chapter presents an overall picture of the story understanding process. It describes what a story understander must accomplish in order to understand a text, and how this goal can be realized. As the process of understanding is examined, the information about intentions required by the process will become more apparent, as will the need to organize this knowledge.

## 2.2 Sample Story

Most of this chapter is concerned with a detailed examination of the problems involved in understanding one simple natural language story. This story serves to illustrate a number of pervasive issues in story understanding.

(1) a) John wanted Bill's bicycle.
b) He walked over to Bill
c) and asked him if he would give it to him.
d) Bill refused.
e) Then John told Bill he would give him five dollars for it,
f) but Bill would not agree.
g) John told Bill he would break his arm if he didn't let him have it.
h) Bill let John have the bicycle.

Q1) Why did John walk over to Bill?
A1) Because he wanted to get his bicycle.

Q2) Why did Bill give his bicycle to John?
A2) Because he didn't want to get hurt.

A trace of PAM understanding this story is shown at the end of this chapter.

## 2.3 Explanation-Driven Understanding

Before I describe my theory of the processing that a reader must perform to understand this text, consider what this process is supposed to accomplish. In the first chapter, I argued that story understanding requires the reader to make a great number of inferences in order to establish the connection between the events in the text. These inferences, together with the explicit events of the story, must be stored by the reader in a memory representation that is amenable for subsequent retrieval tasks. In story (1), the need for making such inferences and including them in the memory representation of the story is demonstrated by the question answering session following the story.

For example, the answer to question Q2 (Why did Bill give his bicycle to John?), is an inference made by the reader. Answering this question involves examining the memory representation of the story to find the event in which Bill gave John the bicycle. Once this event has been found, a retrievel mechanism can check to see if a reason for the event is recorded in the representation. In this case, the reason that is found is the inference, made during story understanding, that Bill gave John the bicycle because Bill didn't want to get hurt.

Making this inference requires that the understander interpret sentence (g) of (1) as an instance of a threat, in particular, as a threat against John's physical well-being. Interpreting (g) as a threat helps explain Bill's behavior in (h). Associated with the concept of

threatening in the memory of the reader is the following information: When a person threatened does the action requested of him, then it can be assumed that he did it in order to avoid the realization of the threat. Thus when a reader encounters sentence (h), he can use this information about threats to infer that Bill gave John the bicycle in order to avoid getting his arm broken by John.

The inference that Bill gave John the bicycle to avoid getting hurt is an example of an explanation. That is, the reader who made this inference must have been engaged in the task of trying to figure out why each event in the story took place. To see what would happen if this were not the case, consider the following version of story (1):

(2) a) John wanted Bill's bicycle.
b) He walked over to Bill
c) and asked him if he would give it to him.
d) Bill refused.
e) Then John told Bill he would give him five dollars for it,
f) but Bill would not agree.
g) John told Bill he would break his arm if he didn't let him have it.
h') Bill ate a banana.

Sentence (h') is difficult to comprehend in the context of story (2). While sentence (h') is a perfectly well-formed and meaningful utterance. the understander cannot easily find an explanation for this event in the situation in which Bill is involved.

Consider what can be done to make sentence (h') acceptable. Suppose we were told that Bill thought eating bananas would make him strong. Or suppose we were told that after he ate the banana, Bill threw the banana peel on the ground and ran away. In the first case, the reader would probably assume that Bill was trying to increase his strength to make him more capable of defending himself against John's threat. In the second case, the reader might infer that Bill was planning an evasive maneuver to get away from John, and was hoping the banana peel would trip John up.

In either of these scenarios sentence (h') seems much more reasonable. What makes it reasonable is that the additional information provided in each scenario enabled the reader to understand why Bill might have performed the action mentioned in sentence (h'). That is, the reader would be able to construct an explanation for the event that he would not be able to find otherwise.

This example serves to demonstrate the role that explanation plays in story understanding. Story understanding is explanation-driven. Much of the processing that a reader needs to do to understand story (1) revolves around the task of finding explanations for the events of the text. In order to build a natural language understander, then, we need to know what these explanatory inferences look like, and how an understander can decide that an explanation has been found.

### 2.3.1 What Is an Explanation?

An explanation often consists of a reason why an actor chose to perform a particular action. Most of the sentences in story (1) have explanations of this sort behind them. For example, John went over to Bill in sentence (b) because that enabled him to try to get Bill's bicycle away from him. In sentence (c), John asked Bill for the bike because asking is a possible way of getting something from somebody, and John wanted Bill's bicycle. Bill's refusal in (d) can be explained by his desire to maintain possession of the bicycle. The explanation for these sentences consist of relating the events to the goals of characters by means of plans.

By the term goal I mean some state of affairs that a character wishes to bring about. For example, satisfying one's hunger can be a goal, as can possessing a bicycle, or being President of the United States.

A plan is some method for achieving a goal. Plans range from the simplistic, like asking somebody for something, to the complex, like running a political campaign.

Both plans and goals have received a great deal of attention in the AI literature. The sense in which I use these terms is taken from Schank and Abelson (1977). I will have more to say about the nature of these items later on. Plans and goals figure prominently in the process of finding an explanation for an event. This process is summarized in Figure 1.

Thus, when an understander reads that John asked Bill to give him his bicycle in story (1), Step 1 of the explanation algorithm checks to see if the understander already knows of a plan of which this event may be a part. Since it knows no such plan, Step 2 of the algorithm tries to infer one. Asking someone for something is a way to get something from that person, so the reader infers that John is using the plan of asking. Then Step 3 looks for a goal to which to attach this plan. Previously in this story, the reader learned that John had the goal of possessing Bill's bicycle. The plan of asking can be used for this goal, so an explanation is established: John did this action because it was a way to fulfill his goal of getting Bill's bicycle.

In story (1), the explanation algorithm stopped after Step 3 because John's goal of wanting Bill's bicycle was already known to the understander. However, consider the following story:

(3) John was hungry. He asked Bill where a restaurant was.

In trying to explain the second sentence of (3), the reader will reach Step 3 of the explanation algorithm, just as it did in the previous example. But here John's plan does not appear to be directly applicable to his stated goal. That is, asking someone the location of something is not a way of becoming less hungry.

Figure 1
The Process of Finding an Explanation for an Event

```
STEP 1                           yes
Is event part of a known         ===>  The plan is the explanation
plan?

                     ||
                     ||  no
                     \/
STEP 2
Can a plan (or plans) be         no
inferred from the event?         ===>  Fail

                     ||
                     ||  yes
                     \/
STEP 3
Can one of these plans be        yes
a plan for a known goal?         ===>  The plan is the explanation
  /\
  ||                 ||
  ||                 ||  no
  ||                 \/
  ||  STEP 4
  ||  Can a goal (or goals) be              no
  ||  inferred from one of these            ===>  Fail
  ||  plans?
  ||
  ||                 ||
  ||                 ||  yes
  ||                 \/
  ||  STEP 5
  ||  Can one of these goals                yes   The plan-goal
  ||  be instrumental to a                  ===>  sequence is the
  ||  known plan?                                 explanation.
  ||
  ||                 ||
  ||                 ||  no
  ||                 \/
  ||  STEP 6
  ||  Can one of these goals                yes   Plan-goal-theme
  ||  have arisen because of a              ===>  sequence is the
  ||  known theme (themes are                     explanation.
  ||  described below)?
  ||
  ||                 ||
  ||                 ||  no
  ||                 \/
  ||       STEP 7
  ||  yes  Can a plan be inferred           no
  \\====   to which one of these            ===>  Fail
           goals is instrumental?
```

To find an explanation for this sentence, most human readers would continue with the following train of thought:

> John probably asked Bill where a restaurant was in order to find out where one was. He might have wanted to know where one was because he wanted to go there. The reason a person usually goes to a restaurant is to get something to eat. But eating is a way to satisfy hunger, which was John's original goal. Thus John must have asked this question because he was trying to find out where a restaurant was in order to go there and get something to eat.

Since eating is a plan for John's already established goal of satisfying his hunger, an explanation has been found.

Thus explaining an event may entail explaining an inferred explanation. One way to explain why a person has a goal is by the principle of <u>instrumentality</u>: A goal often arises because it is instrumental to a plan for another goal. To explain the existence of a goal, try to infer a plan to which achieving that goal may be instrumental. Continue the process until an inference leads to a goal or plan already present in the story representation. This form of explanation is related to Newell and Simon's means-ends analysis (1972). However, rather than reduce a goal to subgoals, this process infers a goal from a subgoal. This process is summarized in steps 3 through 7 of Figure 1.

But not all goals are subgoals. For example, suppose the reader is given the second sentence of (3) without the first, i. e., "John asked Bill where a restaurant was". The reader will go on to infer that John wanted to go to a restaurant because he the goal of satisfying his hungry. But this goal is not instrumental to anything higher.

The inference process stops because hunger is an example of a <u>theme</u>. A theme is a device introduced by Schank and Abelson (1977) to explain the spontaneous occurrence of goals (Schank and Abelson did not include hunger in their classification of themes, but I believe this to be consistent with their usage). Themes themselves are not intentional entities, but are useful in natural language understanding for explaining where an intention comes from.

For example, the hunger theme states that a person becomes hungry after not eating for several hours. If a reader learns that a person wants to eat, the reader will postulate that an explanation for this goal is that the person was hungry. Since hunger is marked as a theme, the reader knows that a complete explanation has been found, and that he has therefore understood the sentence. This part of the explanation process is shown in Step 6 of Figure 1.

The hunger theme is an instance of a class of themes called <u>drives</u> that also includes states like being tired and being sexually aroused. Some other examples of themes are <u>attitudes</u> (see Schank et al, 1978) and <u>social roles</u>. For example, a person who has the attitude of extreme fondness toward someone may have the goal of being with that person. A

person who has the social role of garbage man may have the goal of picking up garbage. In either case, when an understander has learned that a goal arose for one of these reasons, then the understander need not search any further for an explanation. I will discuss themes in more detail in the next chapter.

The explanation process is completed when the meaning of the sentence read can be connected to an already known plan, goal, or theme. If no such item is yet known to the reader, then the process can be stopped when a theme has been inferred from the input. The types of intentional explanations are summarized in Figure 2.

Figure 2
Types of Intentional Explanations

| Item to Explain | Explanation |
|---|---|
| Event | Plan of which event is a part |
| Plan | Goal at which plan is directed |
| Goal | Theme that gave rise to the goal, or plan to which goal is instrumental |

Of course, these intentional forms of explanation are not the only kinds of explanations needed for story understanding. For example, causal explanations are also needed to explain events in the physical world, and to explain automatic reactions (e. g., crying). Causal explanations are discussed in Schank (1973).

## 2.4 Processing Example

In this section I apply the general understanding algorithm sketched above to story (1):

(1) a) John wanted Bill's bicycle.
b) He walked over to Bill
c) and asked him if he would give it to him.
d) Bill refused.
e) Then John told Bill he would give him five dollars for it,
f) but Bill would not agree.
g) John told Bill he would break his arm if he didn't let him have it.
h) Bill let John have the bicycle.

The purpose of this demonstration is to show how the need for finding explanations drives the story understanding process. I will show the kinds of inferences that explanatory understanding forces the reader to make, and suggest processes that can be used to make these inferences.

### 2.4.1 The Input

The algorithms for explanatory processing work with event descriptions. The actual text of story (1) consists of English words, but these are not necessarily the best input for the process of explanation. English words are ambiguous and vague out of context, and so English sentences are poor descriptions of events.

Further, suppose the sentences from story (1) were written in some language other than English. The process of explanation for each sentence would be unchanged. An understander would still need to infer the reason behind Bill's giving John the bicycle in (h), and the reason inferred would still involve Bill's goal of preserving his health. The reason would be the same because what is being explained is the event underlying the words of sentence (h), and not the particular linguistic construct used to express this event. Similarly, if sentence (h) were replaced by an English paraphrase, such as "Bill gave John the bicycle" or "Bill let John take the bicycle", the process of explanation would not be affected by this difference.

The theory of explanation I have been proposing is concerned with processing the content of a text. What needs explaining are the events, goals, states, etc., that are described by the text. The process of explanation is essentially non-linguistic, and therefore should operate on something other than words of a particular language.

To design a system that finds explanations, I needed a language-independent way to express the content of the sentences in a text. I used Schank's Conceptual Dependency theory (Schank, 1975) precisely for this reason. Conceptual Dependency is a system for representing the meanings of natural language utterances. Conceptual dependency structures, called <u>conceptualizations</u>, have the following attractive properties: They are language independent, meaning that they are not biased toward any particular natural language, and they are canonical, meaning that sentences with the same meaning have the same Conceptual Dependency representation.

By assuming that Conceptual Dependency would be the input to my explanation finding process, I was able to specify the explanation process in a language-independent manner. Moreover, this algorithm and the PAM program that implemented it did not have to be concerned with recognizing numerous different linguistic expressions all having the same content, since the Conceptual Dependency representation of these different utterances would all be the same.

Of course, assuming that my algorithms would operate on Conceptual Dependency structures would not be of much use if I had no way of creating such structures from the actual story text. Fortunately, there

is a program that can create these structure from English language sentences. This program, called ELI (English Language Interpreter) was the creation of Riesbeck (1975), and serves as PAM's front end. Whenever I refer to PAM or one of its algorithms as reading a sentence, I mean that my program asked ELI to read the English sentence and that ELI created a Conceptual Dependency representation for it. All my program sees is this representation. The actual story might have been written in Chinese as far as PAM is concerned.

A very brief description of Conceptual Dependency (CD) is included for reference in Appendix 1 of this thesis. Since I make extensive use of CD later on, the reader not familiar with this notation may wish to consult Schank (1975).

2.4.2 The Process

This section demonstrates the explanation algorithm previously presented by applying it to story (1). The description given below is a fair characterization of how PAM actually processes this story. It has been my goal in developing this theory of explanation to simulate a human understander as much as possible. While I have no proof that my simulation is accurate, I intend that it be judged by its psychological appeal as well as by its ability to accomplish a specific natural language understanding task.

An abbreviated version of the CD representation of sentence (a) is the following:

```
 __                         __
|                             |
| BICYCLE IS POSS-BY(JOHN)    |  IS    GOAL( JOHN )
|       |                     |
| POSS-BY( BILL )             |
|__                         __|
```

[(a) John wanted Bill's bicycle.]

This CD encodes the information that John has the goal of possessing a bicycle that is currently owned by Bill.

Having converted the English sentence into CD, the story understander now tries to find an explanation for it. Since this is the first sentence of the story and no previous goals or plans have yet been seen, the reader must be entirely bottom-up. First, the reader must determine what sort of input it is, e. g., whether it is an event or a state, since different inputs require different types of explanation. By virtue of the appearance of the "IS GOAL" portion of the CD, the reader is able to determine that this CD is an instance of a goal.

Goals can be explained either by instrumentality or by themes. Thus the reader must see if it knows of any themes that can give rise to this particular goal, or of any plans for which this goal would be a subgoal. The goal of possessing any object can arise from the theme of liking that object. Also, possessing an object with a standard function

can be instrumental to using that object. The reader should therefore be able to infer two plausible explanations for this sentence: One, that John wanted to have Bill's bicycle simply because he liked it, and two, that John may have wanted to have Bill's bicycle because he wanted to ride it.

In order to produce these explanations, the reader needed to know something about wanting to own a bicycle, and had to be able to retrieve this information upon encountering the input CD. What the reader needed to know in this case can be stated as follows:

(1) If a person wants to possess an object, then he may have the attitude of liking that object.

(2) If a person wants to possess an object that has a function, then he may want to use that object to perform its function.

These are examples of knowledge expressed as <u>rules</u>. A rule consists of two parts, a condition and an action. If the condition of a rule is met, then the action of that rule is taken. The action of a rule is to make an inference. For example, when rule (1) is applied to the input CD for (a), its condition (that a person wants to own an object) is met, and the action generates the inference that John may be fond of the bicycle. The condition of the second rule is also satisfied by the input, and it produces the inference that John may want to ride the bicycle.

The process of making an inference from an input CD consists of finding those rules whose tests are met by the CD, and then taking the actions expressed in these rules. To determine if the condition of a rule is met, the reader needs access to a wide variety of knowledge about the world. Much of the later part of this thesis is concerned with this knowledge and how it is used in story understanding. For example, the second rule requires the reader to know that a bicycle has a use, and that its use is to be ridden. This is an instance of knowledge about <u>objects</u>. A discussion of the role object knowledge plays in story understanding is found in Chapter 11.

Having accessed the rules whose condition is met by the input CD, and having made the inferences indicated by the actions of these rules, the understander is left with two alternative explanations for the sentence. At the beginning of the understanding process, the reader has no way of determining which of these explanations is the correct one. However, since the understander has determined several plausible explanations, it need not continue inferring. Instead, the understander merely records both plausible inferences in the story representation, and hopes that a future event in the story will help decide between them.

So after processing the first sentence of (1), the story representation built up so far looks like this:

Story Representation for Story (1) After Sentence (a)

```
 John is fond of bicycle    John wants to ride bicycle
            \                        /
             \                      /
              \                    /<-- plausible sources
               \<--------------/----/
                \              /

        Goal - John wants to possess Bill's Bicycle.
                  Outcome - (unknown)
```

While this figure is expressed in English for convenience, the actual representation is built out of Conceptual Dependencies and similar structures. This diagram shows that the CD representation for the input sentence (a) has been added to the memory representation along with two potential <u>sources</u>. A source is a general term I use for anything that can give rise to a goal. Sources include both themes and plans to which the goal may be instrumental.

The understander now reads sentence (b) and converts it into CD:

```
                                         |-> PROX( BILL )
JOHN <=> PTRANS <--O-- JOHN --|
          ^                              |-<
          |
          | I
          |
JOHN <=> MOVE <--O-- LEGS
```

[(b) He walked over to Bill.]

That is, John changed his location to be near Bill by moving his legs.

Note that I took the liberty of replacing the pronoun "he" with its referent "John" in the CD. The details of this pronoun replacement scheme are described in Chapter 13. For the sake of brevity, the issue of reference is ignored in the remainder of this discussion.

The reader now tries to find an explanation for this CD. Because the CD contains the action designator "<=>", the reader knows that this is an event. According to the theory of explanation expounded above, the explanation of an event is a plan underlying it. Since no plan is currently known in the story, the algorithm sketched in the previous section suggests that the reader try to infer one.

Once again, the reader tries to access those rules in memory whose conditions are satisfied by the input. In this case, the following rule is found:

If an actor does a PTRANS whose instrument is the actor moving
    a bodypart,
then that actor is using the plan of moving under his own
    power.

That is, one way of changing one's location is through the use of one's own body. Since this is indicated by the instrument in the CD representation, John must be using this plan.

Now the reader tries to connect this inferred plan with the goals present in the story representation built so far. The only goal in this representation is John's desire for Bill's bicycle. Thus the understander checks to see if going over to someone is a plan for this goal. The understander examines the plans it knows about for gaining possession, and concludes that a plan for moving about under one's own steam is not among them. More of an explanation is necessary.

According to the theory of explanation, the reader needs a goal to which this plan can be connected. The reader therefore tries to infer the goal behind John's plan. By accessing and applying rules as above, the reader infers that John wanted to be near Bill. Now the reader checks to see if this goal can be connected to the story representation. Associated with the goal of getting something from someone is the fact that being near someone is often instrumental to the plans for that goal. Since the first goal is already in the story representation, the reader can infer that John had the second goal because it is instrumental to the first via some unknown plan. Hence an explanation has been found, and the inferences made in order to find this explanation are added to the story representation:

Story (1) After Sentence (b)
(The box in the diagram represents new material.)

```
 Theme - John is likes bicycle   Plan - John wants to ride bicycle
          \                      /
           \                    /<-- plausible sources
            \<----------------/----/
             \               /
       Goal - John wants to possess Bill's Bicycle.
                  Outcome - (unknown)
  ______________________|___________________________________
 |                      |                                   |
 | Plan - (unknown)                                         |
 |                      |                                   |
 |                      |                                   |
 | Goal - John wants to be near Bill.                       |
 |        Outcome - (fulfilled)                             |
 |                      |                                   |
 |                      |                                   |
 | Plan - Use own power  --->  Action - John walk to Bill   |
 |__________________________________________________________|
```

The reader has added the fact that John had the goal of being near Bill,

that this was instrumental to some plan for his goal of getting Bill's bike, and that John succeeded in getting near Bill by transporting himself over to him.

Now sentence (c) - "and asked him if he would give it to him." - is read and converted to CD, and an attempt is made to explain it. Sentence (c) is also an action, so the reader tries to infer a plan from it. The reader infers that sentence (c) meets the criteria of the plan of asking someone for something, and tries to find a goal in the representation to which this plan may be attached.

Stored in the reader's memory along with the goal of getting something from someone is a list of possible plans that includes asking. Thus this plan must be the unknown plan John intended to use when he walked over to Bill. Sentence (c) is thereby explained. The story representation now looks like this:

Story (1) After Sentence (c)

```
Theme - John is likes bicycle    Plan - John wants to ride bicycle
          \                         /
           \                       /
            \                     /<----plausible sources
             \<-------------/-----/
              \            /
        Goal - John wants to possess Bill's Bicycle.
               Outcome - (unknown)
                    |
 ___________________|_______________________________________
|                   |                                       |
| Plan - Asking     |          ----> Action - John ask Bill |
|___________________|_______________________________________|
                    |
                    |
        Instrumental Goal - John wants to be near Bill.
               Outcome - (fulfilled)
                    |
                    |
                    |
        Plan - Use own power    ----> Action - John walk
```

The top-down element in the understanding process is provided by the rules of explanation that require the understander to connect structures inferred from the input to structures already in the story representation. For example, when the reader infers a plan like asking, the reader then looks for a goal in the representation to which that plan can be attached. If a goal is found, as it was above, the explanation process stops. If no goal is found, as was the case in example (4), the reader would try to infer a goal from the inferred plan.

The existence of a goal in the representation therefore serves a predictive function. It predicts that a plan for that goal may be inferred from the input. For example, the existence of John's goal of possessing Bill's bicycle caused the reader to predict that a plan for this goal would be encountered. When the plan of asking was inferred from the input, this prediction was confirmed because asking could indeed be a way of getting an object from someone.

The concept of prediction provides a convenient way to describe and implement the explanation algorithm. When a sentence is encountered, explanations are inferred from the sentence until one of these explanations confirms one of the existing predictions. These predictions were made as the result of processing a previous sentence. For example, the existence of a goal without a plan creates the prediction that a plan will be seen; a plan without an action predicts an action; and a theme without a goal predicts a goal.

These are predictions about the likelihood that some knowledge will be useful for understanding a story. They are not necessarily predictions about future events. For example, suppose John asks Bill a question. One way Bill might respond is by answering the question. When I say this response is predicted by the asking event, I mean that a prediction is made that the following knowledge will be useful: "If Bill tells John what Bill was trying to find out, then Bill told this to John in response to John's question." This is not a prediction that Bill will answer the question, or that he is even likely to answer it. Rather, it is a prediction that knowledge about responding will be useful to understand subsequent sentences of the story. This sense of prediction is similar to that used by Kuno and Oettinger (1962) and by Riesbeck (1975).

Creating a prediction is implemented by setting up a demon-like routine (see Charniak (1972), and Selfridge (1959)) that examines each subsequent input and each inference made from them. These routines are also similar to the requests used in Riesbeck's conceptual analysis program (see Riesbeck, 1975). A request is a kind of rule in that it has a condition part and an action part. If the condition part is met by an input or an inference from an input, the action part will connect the input up to the story representation.

For example, upon seeing the goal in sentence (a) of story (1), the understander make a prediction that it would see a plan for this goal. When the plan of asking is inferred upon reading sentence (c), the condition of the prediction is met, and the action of the prediction will connect the inferred plan to the representation for John's "possess bicycle" goal.

It is often possible to generate explanatory predictions when the goal in a story cannot be fully specified. For example, after sentence (c), Bill's subsequent behavior is likely to be in response to John's question. If he gives John the bicycle, the reader should infer that it was because John asked him for it. If he refuses, then the reader should assume he is refusing John's request. However, before the reader sees sentence (d), the reader cannot know what Bill's reaction is going to be.

Therefore, the reader sets up two predictions, one for each possible outcome. When sentence (d) is encountered, the prediction looking for a denial of the request is confirmed, and an explanation of Bill's action created. The representation for the story then looks like this:

Story (1) After Sentence (d)

```
      .                           .
      .                           .
      .                           .

Goal - John wants to possess Bill's Bicycle.
       Outcome - (unknown)
              |
              |                   ____________________________
              |                  |                            |
Plan - Asking  ----> Action ----> Theme - Preservation        |
      .         (John ask Bill)|      |                       |
      .                        |      |                       |
      .                    ____|      |                       |
                          |           |                       |
                          | Goal - Bill wants to              |
                          |           keep bicycle            |
                          |           Outcome - (unknown)     |
                          |               |                   |
                __________|               |                   |
               |                                              |
               |   Action  <---- Plan - DENY                  |
               | (Bill refuse)                                |
               |______________________________________________|
```

That is, John's plan caused Bill to have the goal of keeping possession of the bicycle by virtue of the theme that people like to keep things they already own. Note that according to the use of the term goal here, John did not have the goal of keeping his bicycle until his possession of it was threatened. In general, I consider the maintenance or preservation of a state to be a goal only when a character must do some action on its behalf. This terminology avoids the uncomfortable position of having to say that John always has the goal of preserving his bicycle. Instead, John is always operating under a theme that gives rise to particular preservation goals as the need arises. Preservation goals are discussed in Chapter 3.

The top-down nature of prediction was useful here because it enabled the understander to interpret sentence (d) as a refusal of John's request. The actual sentence, however, does not mention what Bill was refusing, so it would have been difficult to make this inference in a strictly bottom-up system, i.e., one that made all its inferences directly from the input.

Predictions are also used to find explanations for the other sentences of story (1). For example, when John's plan fails in (d), a new prediction is added looking for a new plan for the same goal. When

sentence (e) is read, it is inferred to be an instance of the plan of bargaining. Since this plan can be used for John's goal, the new prediction is fulfilled, and the event is explained. Predictions are then set up which are similar to those made after sentence (c), looking for the different possible outcomes. Sentence (f) confirms the prediction that Bill might decline the deal. The confirmation of this prediction causes another prediction to be set up looking for yet another plan.

Sentence (g) confirms this prediction because the bottom-up inference process infers that it is an instance of the plan of threatening, also a way to get something from someone. Predictions are set up again looking for possible outcomes. One of the predictions set up by the plan of threatening is that if the person threatened does the action requested of him, then he does it to avoid the threat happening. Thus when sentence (h) is read, it is this prediction that is confirmed. The prediction explains sentence (h) by building the interpretation that Bill must have wanted to maintain his health, and therefore complied with John's demand. The representation of the story now includes the following:

Story (1) After Sentence (h)

```
            .                         .
             .                       .
            .                       .
     Goal - John wants to possess Bill's Bicycle.
            Outcome - (fulfilled)
                  |       |         |
                  |       |         |
     Plan - Asking...     |         |
         .                |         |
         .   Plan - Bargaining ...  |
         .                          |
                                    |
               Plan - Threatening ---> Action - John tells Bill
                                                |
                                                |
                                     ___________|___________
                                    |          \/           |
                                    | Theme - Preservation  |
                                    |            |          |
                                    |            |          |
                                    | Goal - Maintain health |
      ______________________________|            |          |
     |                                           |          |
     | Action - Bill gives John bike  <---  Plan - Complying |
     |______________________________________________________|
```

Here John's plan induced a conflict with Bill's previous desire, so Bill chose to comply with John's request by giving John the bicycle. This fulfilled John's original goal of having Bill's bicycle, and the goal is marked as such in the representation.

## 2.5 Computer Example

This section consists of an annotated PAM run on story (1). Since the actual plans and goals known to PAM have not yet been specified, the following trace of PAM's processing may be somewhat cryptic. It is included here to give some indication of how the explanation algorithm can be implemented. The plans and goals used by PAM are discussed in the next chapter. Chapters 13 and 14 provide a detailed explanation of how the program works.

```
[PHOTO: Recording initiated  Thu 30-Mar-78 4:29PM]

@RUN PAM

*(UNDERSTAND CD6)

THE STORY IS

 JOHN WANTED BILL'S BICYCLE.
 HE WENT OVER TO BILL
  AND ASKED HIM IF HE WOULD GIVE IT TO HIM.
 BILL REFUSED.
 JOHN TOLD BILL HE WOULD GIVE HIM FIVE DOLLARS FOR IT,
  BUT BILL WOULD NOT AGREE.
 THEN JOHN TOLD BILL HE WOULD BREAK HIS ARM
  IF HE DIDN'T LET HIM HAVE IT.
 BILL GAVE HIM THE BICYCLE.
```

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(JOHN WANTED BILL'S BICYCLE) | |
| CONCEPTUALIZATION IS:<br>((CON<br>((ACTOR PHYS0 IS<br>(*POSS* PART HUM0))<br>TIME (FORM11))<br>IS (*GOAL* PART HUM0))<br>TIME (FORM12)) | PHYS0 is a token (representation for an individual entity) created by PAM for the bicycle. HUM0 is a token for John, and HUM1 a token for Bill. The time forms contain information about time predications in an internal format. For example, FORM12 states that the entire event occurred in the past, and FORM11 that John having the bicycle may exist in the future. |
| NOT A PREDICTED INPUT<br><br>BEGIN SEARCH FOR EXPLANATION | Since no prediction is available to explain the input, PAM uses its bottom-up mechanism to find an applicable rule. |

| | |
|---|---|
| TESTING EXPLANATION OFFERED BY<br>DCONT-EPISODE-REQ | This is the name of the rule found (recall that rules in PAM are called requests). |
| | DCONT denotes a "change of control" goal. Entities like DCONT are discussed in the Chapter 3. |
| EXPLANATION IS GOAL:<br>(*DCONT* PLANNER HUM0 OBJECT PHYS0<br>OWNER HUM1 RECIPIENT HUM0) | The rule found states that John has the goal of taking possession of the bicycle. Now an explanation for this goal is sought. |
| NO PREDICTION CONFIRMED | |
| ASSUMING EXPLANATION<br>CONTINUING SEARCH | |
| TESTING EXPLANATION OFFERED BY<br>USE-VEHICLE-REQ FONDNESS-REQ | PAM finds that the goal could have come from liking the object or wanting to use it to get someplace. |
| EXPLANATION IS PREDICTED PLAN:<br>*PB-USE-VEHICLE* | This is the name of a plan for changing one's location, |
| EXPLANATION IS PREDICTED THEME:<br>*FONDNESS* | and this is a theme that represents liking something. |
| EXPLANATION CONFIRMS PREDICTION<br>INIT-REQ<br><br>FOUND EXPLANATION SEQUENCE:<br><br>DCONT-EPISODE-REQ -> USE-VEHICLE-REQ | INIT-REQ is a prediction made at the start of a story that looks for a goal with plausible but uncertain sources. This rule is a way of letting PAM know when to stop processing the initial sentence of a story. The rules used to find the explanation are now used to actually connect the input to the representation. |
| *** ADDING TO STORY REPRESENTATION: | |
| LOADING PREDICTION<br>DCONT-ATTEMPT-REQ | This predictions looks for a plan for John's goal to appear in the story. |
| GOAL: (*DCONT* PLANNER HUM0 OBJECT<br>PHYS0 OWNER HUM1 RECIPIENT HUM0) | |
| NEXT INPUT IS:<br>(HE WENT OVER TO BILL) | |

```
CONCEPTUALIZATION IS:
((ACTOR HUM2 <=> (*PTRANS*)
  OBJECT HUM2 TO (*PROX* PART HUM1)
  INST (NIL))
 TIME (FORM17))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
WALK-GOAL-EPISODE-REQ
```

Again, PAM goes bottom-up to find an explanation. PAM finds a rule that it uses to infer that John wanted to be near Bill to do some plan to get his bicycle. The rest of the processing of this sentence has been deleted from the story trace for the sake of brevity.

```
EXPLANATION IS GOAL:
(*DPROX* PLANNER HUM2 OBJECT HUM2
 LOCATION (*PROX* PART HUM1))
 PLAN: (*PB-WALK* PLANNER HUM2
LOCATION (*PROX* PART HUM1))
                .
                .
                .

NEXT INPUT IS:
 (AND ASKED HIM IF HE WOULD GIVE IT
  TO HIM)

CONCEPTUALIZATION IS:
((ACTOR HUM3 <=> (*MTRANS*)
  MOBJECT
  ((CON
    ((ACTOR HUM4 <=> (*ATRANS*)
      OBJECT PHYS0 TO HUM3)
     TIME (FORM33))
    LEADTO
   ((ACTOR HUM3
     TOWARD (*JOY* VAL (NIL)))
     INC (2.) TIME (FORM34))))
  TO (*CP* PART HUM4))
 TIME (FORM35))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
ASK-REQ
```

PAM finds a rule interpreting the input as part of the plan of asking.

```
EXPLANATION IS
 PLAN: (*PB-ASK* PLANNER HUM3
PERSUADEE HUM4 REQUEST
((ACTOR HUM4 <=> (*ATRANS*) OBJECT
  PHYS0 TO HUM3) TIME (TIMK5)))


EXPLANATION CONFIRMS PREDICTION
DCONT-ATTEMPT-REQ

FOUND EXPLANATION SEQUENCE:

ASK-REQ
```

The prediction set up after reading the first sentence recognizes this as a plan for the goal of getting some money, so the input is explained.

```
*** ADDING TO STORY REPRESENTATION:

INFERRED
 PLAN: (*PB-ASK* PLANNER HUM0
PERSUADEE HUM1 REQUEST
((ACTOR HUM1 <=> (*ATRANS*) OBJECT
  PHYS0 TO HUM0) TIME (TIMK5)))


LOADING PREDICTION
COMPLIANCE-GOAL-REQ
```

Predictions are loaded for the possible responses to John's plan.

```
LOADING PREDICTION
REFUSAL-GOAL-REQ

NEXT INPUT IS:
 (BILL REFUSED)

CONCEPTUALIZATION IS:
((ACTOR HUM1 <=> (*MTRANS*)
  MOBJECT
 ((ACTOR HUM1 <=> (*DO*))
  MODE (FORM45) TIME (FORM46))
 TO (*CP* PART HUM5))
 TIME (FORM47))


NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
TELL-REQ
```

PAM infers that Bill was trying to tell John something,

```
EXPLANATION IS
 PLAN: (*PB-TELL* PLANNER HUM1
TELLEE HUM0 FACT
((ACTOR HUM1 <=> (*ATRANS*)
  OBJECT PHYS0 TO HUM0)
 MODE (FORM52) TIME (TIMK5)))
```

```
NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
INFORM-DECISION-REQ

EXPLANATION IS GOAL:
(*DKNOW* PLANNER HUM1
RECIPIENT HUMO FACT
 ((ACTOR HUM1 <=> (*ATRANS*)
   OBJECT PHYSO TO HUMO)
  MODE FORM52) TIME (TIMK5)))
 PLAN: (*PB-TELL* PLANNER
HUM1 TELLEE HUMO FACT
 ((ACTOR HUM1 <=> (*ATRANS*)
   OBJECT PHYSO TO HUMO)
  MODE (FORM52) TIME (TIMK5)))
```

and that his goal was to inform John of the decision he had reached.

```
EXPLANATION CONFIRMS PREDICTION
REFUSAL-GOAL-REQ

FOUND EXPLANATION SEQUENCE:

TELL-REQ -> INFORM-DECISION-REQ
```

This inferences confirms the prediction that Bill might refuse the request, so an explanation for the input has been found.

```
*** ADDING TO STORY REPRESENTATION:

INFERRED
 PLAN: (*PB-TELL* PLANNER HUM1
TELLEE HUMO FACT
((ACTOR HUM1 <=> (*ATRANS*) OBJECT
PHYSO TO HUMO) MODE (FORM52)
TIME (TIMK5)))

 GOAL: (*DKNOW* PLANNER HUM1
RECIPIENT HUMO FACT
((ACTOR HUM1 <=> (*ATRANS*) OBJECT
PHYSO TO HUMO) MODE (FORM52)
TIME (TIMK5)))

INPUT CONFIRMS PREDICTION
REFUSAL-GOAL-REQ

INFERRED GOAL: (*PCONT* PLANNER
HUM1 OWNER HUM1 OBJECT PHYSO)
```

The prediction that interpreted the input also adds to the representation that John wanted to keep the bicycle,

```
INFERRED
 PLAN: (*PB-REFUSE* PLANNER HUM1
RECIPIENT HUMO FACT
((ACTOR HUM1 <=> (*ATRANS*) OBJECT
PHYSO TO HUMO) MODE (FORM52)
TIME (TIMK5)))
```

and that he did so by refusing John's request.

```
NEXT INPUT IS:
 (JOHN TOLD BILL HE WOULD GIVE HIM
  FIVE DOLLARS FOR IT)
```

```
CONCEPTUALIZATION IS:
((ACTOR HUMO <=> (*MTRANS*)
  MOBJECT
  ((CON
    ((ACTOR HUM6 <=> (*ATRANS*)
      OBJECT PHYSO TO HUMO)
     TIME (FORM66))
    LEADTO
    ((ACTOR HUMO <=> (*ATRANS*)
      OBJECT MONEY1 TO HUM6)
     TIME (FORM67))))
  TO (*CP* PART HUM6))
 TIME (FORM68))
           .
           .
           .
```

John tries bargaining. This sent-sentence and the next sentence are processed almost identically to the last two, so their processing will be skipped over here.

```
NEXT INPUT IS:
((ACTOR HUMO <=> (*MTRANS*)
  MOBJECT
  ((CON
   ((ACTOR HUM1 <=> (*ATRANS*)
     OBJECT PHYSO TO HUM1)
    MODE (FORM99) TIME (FORM100))
   LEADTO
   ((CON
    ((ACTOR HUMO <=> (*DO*))
     TIME (FORM101))
    LEADTO
    ((ACTOR PARTO
      TOWARD (*PSTATE* VAL (NIL)))
     INC (-5.) TIME (FORM102))))))
  TO (*CP* PART HUM1))
 TIME (FORM103))
```

*NOT A PREDICTED INPUT*

```
BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
THREATEN-REQ
```

PAM identifies this action as a threat,

```
EXPLANATION IS
 PLAN: (*PB-THREATEN* PLANNER HUM0
REQUEST
((ACTOR HUM1 <=> (*ATRANS*)
  OBJECT PHYS0 TO HUM1)
 MODE (FORM110) TIME (TIMK15))
PERSUADEE HUM1 THREAT
((CON
  ((ACTOR HUM0 <=> (*DO*))
   TIME (TIMK16))
  LEADTO
  ((ACTOR PART0 TOWARD
   (*PSTATE* VAL (NIL)))
   INC (-5.) TIME (TIMK17)))))

EXPLANATION CONFIRMS PREDICTION      | which is identified as another
DCONT-ATTEMPT-REQ                    | plan to get the bicycle.

FOUND EXPLANATION SEQUENCE:

THREATEN-REQ

*** ADDING TO STORY REPRESENTATION:

INFERRED
 PLAN: (*PB-THREATEN* PLANNER HUM0
REQUEST
((ACTOR HUM1 <=> (*ATRANS*)
  OBJECT PHYS0 TO HUM1)
 MODE (FORM113) TIME (TIMK15))
PERSUADEE HUM1 THREAT
((CON
  ((ACTOR HUM0 <=> (*DO*))
   TIME (TIMK16))
  LEADTO
  ((ACTOR PART0 TOWARD
    (*PSTATE* VAL (NIL)))
   INC (-5.) TIME (TIMK17)))))

LOADING PREDICTION THR-COMPLY-REQ    |  Predictions are loaded for
THR-COMPLY-REQ                       | possible responses.

LOADING PREDICTION EVADE-THREAT-REQ

LOADING PREDICTION
RESULT-IN-DCONT-REQ
```

```
NEXT INPUT IS:
 (BILL GAVE HIM THE BICYCLE)

CONCEPTUALIZATION IS:
((ACTOR HUM1 <=> (*ATRANS*) OBJECT
  PHYSO TO HUM8) TIME (FORM115))

INPUT CONFIRMS PREDICTION
RESULT-IN-DCONT-REQ
 OUTCOME OF GOAL: (*DCONT* PLANNER
HUMO OBJECT PHYSO OWNER HUM1
RECIPIENT HUMO) IS (*SUCCEED*)
```

PAM recognized that John's goal has been fulfilled, but it still must find an explanation for Bill's action.

```
NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
GIVE-REQ

EXPLANATION IS
 PLAN: (*PB-GIVE* PLANNER HUM1
RECIPIENT HUMO OBJECT PHYSO)

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
GIVE-GOAL-REQ
```

PAM infers that Bill must have wanted John to have the bicycle, and tries to explain this goal.

```
EXPLANATION IS GOAL:
(*DCONT* PLANNER HUM1 OBJECT PHYSO
 OWNER (NIL) RECIPIENT HUMO)
 PLAN: (*PB-GIVE* PLANNER HUM1
RECIPIENT HUMO OBJECT PHYSO)

EXPLANATION CONFIRMS PREDICTION
THR-COMPLY-REQ
```

The prediction stating that Bill might give in to John is confirmed by this goal.

```
FOUND EXPLANATION SEQUENCE:

GIVE-REQ -> GIVE-GOAL-REQ

*** ADDING TO STORY REPRESENTATION:

 INFERRED GOAL: (*DCONT* PLANNER
HUM1 OBJECT PHYSO OWNER (NIL)
RECIPIENT HUMO)
```

| | |
|---|---|
| OUTCOME OF GOAL: (*DCONT* PLANNER HUM1 OBJECT PHYSC OWNER (NIL) RECIPIENT HUMC) IS (*SUCCEED*) | |
| INFERRED GOAL: (*PHEALTH* PLANNER HUM1 RECIPIENT HUM1)<br><br>INFERRING CONFLICT WITH GOAL: (*PCONT* PLANNER HUM1 OWNER HUM1 OBJECT PHYSC) | PAM adds to the representation that Bill wanted to preserve his health, and that this goal conflicted with his goal of keeping the bicycle. |
| FINISHED UNDERSTANDING PHASE | |

```
**********************************************************************
*                                                                    *
*  Some question-answering and paraphrasing follow to demonstrate the *
* level of understanding achieved.                                   *
*                                                                    *
**********************************************************************
```

*(DOQA)

QUESTION: C1
Why did John walk over to Bill?
Because he wanted to get his bicycle.

QUESTION: C2

Why did Bill give his bicycle to John?
Because he didn't want to get hurt.

QUESTION: C3
What were the consequences of John's walking over to Bill?
This enabled him to ask him to give him Bill's bicycle.

QUESTION: C4
What were the consequences of John's asking Bill to give him Bill's bicycle?
Bill told him that Bill wouldn't give him Bill's bicycle.

```
************************************************************************
*                                                                      *
* Expressing the story from different points of view ....              *
*                                                                      *
************************************************************************
```

*(TELL)

WHO SHOULD TELL THE STORY? *JOHN

I wanted to get Bill's bicycle. So I walked over to him, and I asked him to hand it over. He told me that he wouldn't hand it over. So I asked him to sell it to me for five dollars. Then he told me that he wouldn't hand over his bicycle. I told him that if he didn't hand it over then I would break his arm. He handed over his bicycle.

WHO SHOULD TELL THE STORY? BILL

John came over. He asked me to give him my bicycle. I wanted to keep the bicycle, so I told him that I wouldn't give it to him. Then he offered to buy it for five bucks. I wanted to keep the bicycle, so I told him that I wouldn't give it to him. He told me that if I didn't give it to him then he would break my arm. I didn't want to get hurt. So I gave him my bicycle.

[PHOTO: Recording terminated Thu 30-Mar-78 4:34PM]

## 2.6 Summary

The purpose of this chapter was to demonstrate the role intentionality plays in story understanding. A story understander has the job of finding explanations for the sentences of a text. To construct an explanation, it is necessary to postulate explanatory devices called goals, plans, and themes. These are needed to describe and explain the reasons why a character might have performed a particular action, or why he may have desired a particular state of affairs.

To apply these intentional entities to the task of finding the explanation for a sentence, a combination of bottom-up and top-down processing was needed. The bottom-up part of the process consists of finding rules relevant to a given sentence and making the inferences suggested by those rules. The top-down part of the process consists of comparing these inferences against previously made predictions. When a prediction is confirmed by a bottom-up inference, an explanation is found and the inference process is stopped. The explanation can then be incorporated into the representation of the story in memory.

In the previous sections I alluded to some of the difficulties that this theory of explanation leaves us with. In particular, the explanation process presumes that the story understander has a large

body of knowledge about plans, goals, themes and so on, but I did not state what kinds of plans, goals and themes there were, or what these objects looked like in detail. The nature of these items is the subject of the next chapter.

# CHAPTER 3

## Theory of Intentionality

## or, Plans, Goals, Themes, and Understanding

### 3.1 Introduction

In the previous chapter I introduced the notion of explanation to describe the story understander's task: A reader must explain why each event in a text occurred. To facilitate this job, plans, goals, and themes were used as explanatory devices. Plans explain events by relating them to goals, and themes explain where goals came from in the first place.

A natural language understander that can find explanations for the events in a story must embody a theory of intentionality. That is, for an understander to recognize the plans, goals and themes underlying a particular utterance, the understander needs to know what kinds of plans, goals and themes there are in the first place. For example, consider the inferences needed to find an explanation for the sentence

(1) John told Bill he would give him five dollars if Bill gave him his bicycle.

The following chain of inferences needs to be constructed to explain this sentence:

1. Infer the plan - John was using the plan of bargaining.

2. Infer the goal - This plan was used to try to get Bill's bicycle.

3. Infer the source of the goal - The goal may have arisen because John liked Bill's bicycle, or because he wanted to ride the bicycle someplace.

To determine that John was using the plan of bargaining with someone in sentence (1), the reader needs to know that bargaining was in fact a plan. That is, that one particular method of getting something from someone is to offer that person something he might want in exchange for him giving you something you want. Sentence (1) can then be recognized as an instance of bargaining because it meets this criterion: John offered to give Bill money, which is usually deemed desirable, in exchange for Bill giving John the bicycle.

To infer that John had the goal of possessing Bill's bicycle requires that the reader knows what kind of goals people are likely to have. For example, John may have been someone who loved trading, and who merely wanted to engage in a trading transaction rather than end up having a bicycle. While this interpretation is possible, it is not very likely. It is unlikely because wanting to possess an object is a much more common goal. Most readers will use the knowledge that this goal is common to infer the possession goal upon encountering John's trading plan without enumerating all the possible goals this action might conceivably have been directed at.

Inferring that John may have wanted to have the bicycle because he likes it requires that the understander have specific knowledge about liking: The reader must know that a standard reason why a person wants to possess an object is that he likes the object. That is, the reader needs to know that liking something is a theme that can give rise to the goal of possessing that thing.

For each of these inferences, the reader needs to have knowledge about a particular plan, goal or theme. For sentence (1), the reader needs to know about bargaining; he needs to know about the goal of possessing something to infer that this goal underlies the bargaining plan; and he needs to know about the theme of liking something to infer that this theme may have given rise to John's goal.

Thus to find explanations for the sentences of a text, a natural language understander needs knowledge about the set of goals, plans and themes that account for the behavior of a story's characters. This chapter presents such a theory of intentionality. The purpose of this theory is to enumerate and describe the intentional structures needed for natural language understanding.

### 3.2 Background - AI theories of planning.

The concepts of plans and goals are by no means new ideas in Artificial Intelligence. A great deal of work has been done with the use of plans for problem solving and for robot planning. It would certainly seem feasible, then, that the plans and goals used in AI theories of problem solving or the like could be used directly or with some modification as the basis for a plan-based natural language story understander.

Unfortunately, this turned out not to be the case. The tasks these robot problem solvers were meant to perform and the tasks with which a natural language processor is faced are very different. This difference is great enough to merit the construction of a new theoretical apparatus for planning. In this section, I discuss some of the notions of planning that were developed for other AI systems, and show why they were not amenable to the problems I faced.

### 3.2.1 Problem Solvers

Most early AI theories of planning were used in problem solving. For example, one of the earliest problem solvers, the General Problem Solver (GPS) (Newell, Shaw and Simon, 1959) used the notion of a goal to state a problem their program was to solve: A goal was a description of a desired state of the world; a problem was a goal to be achieved given a different state of the world to begin with.

To solve a problem, GPS worked backwards from the goal description. It tried to find a set of operators that could bring about the goal, and then tried to establish the conditions under which these operators could be applied. In this manner, a goal would give rise to a number of subgoals, each presumably easier to achieve than the original goal. Eventually, operators would be found that could be applied to a subgoal in the existing state of the world. These operators were then used to change the state of the world, enabling other operators to be applied until an operator that could effect the original goal was enabled. This technique of problem solving was termed means-end analysis.

GPS was rather non-committal about specifying a particular set of operators. It was designed to explore problem solving at the most general level. Its creators were more interested in elucidating the general techniques of problem solving than they were in codifying specific problem solving knowledge. Thus no particular set of plans emerged from work on GPS that would fulfill our needs here.

However, means-end analysis was incorporated into a number of robot problem solving programs that were directed at particular domains. STRIPS (Fikes and Nilsson, 1971), a problem solver developed at SRI, uses a means-end analysis technique to solve problems in a micro-world consisting of a robot, some boxes, and a few rooms. The robot has the ability to move and to push the boxes around. It is given problems to solve like getting two boxes that are in different rooms to touch one another.

STRIPS solves such problems by representing its world in terms of predicate calculus notation, and then using a resolution theorem prover to determine what needed to be done. For example, to represent the fact that a particular box was in location X, STRIPS would use the predicate AT(BOX1,X). The operations that the robot could perform, MOVE-ROBOT and PUSH-BOX, would cause predicates such as this to be added or removed from the STRIPS data base, depending upon whether the operation made them true or false.

Each operation also has preconditions that must be met in order for it to be used. The robot has to be at the location of a box in order to push it, for example. The STRIPS theorem prover is used to compare the current state of the world with the goal state, and find the differences if there are any. Eliminating these differences then generates subgoals that are operated upon as in GPS.

A great deal of work was done on extending, generalizing, and improving STRIPS. In particular, an enhancement of STRIPS known as ABSTRIPS (Sacerdoti, 1974) organized problems into levels called abstraction spaces. Abstraction spaces are successively more abstract descriptions of the origin problem in which unimportant details are ignored. A problem could be solved more easily on an abstract level, and then a more detailed solution found by expanding the abstract solution at a lower, less abstract level, until a solution was found for the original problem.

There are a great many other problem solving systems. For example, HACKER (Sussman, 1975), NOAH (Sacerdoti, 1977), and NASL (McDermott 1977) represent a number of different approaches. However, it is the general orientation of problem solvers rather than their particular details that is of interest here. For the purpose of comparing problem solvers with PAM, the differences between these systems are not significant.

### 3.2.2 The Problem with Problem Solvers

While these problem solving programs functioned admirably in their domains, their notion of planning was considerably different from the one I found I needed for building a language understanding system. One difference between a problem solver like STRIPS and a story understander like PAM lies in the nature of the task to be performed. STRIPS constructs plans. PAM understands them. That is, PAM does not solve problems; it infers the solutions to problems that characters in stories are using. This is the difference between understanding and generation. PAM could no more solve a problem than a FORTRAN compiler could write a FORTRAN program. Likewise, using STRIPS to understand a story would be like using a line printer as an input device to a computer.

The difference between problem solving and plan understanding manifests itself mostly in the control structure of the program rather than in the theory of planning upon which the program is based. Thus the difference between understanding and generation does not constitute the major drawback in using a STRIPS-like planning system for language understanding. A STRIPS theory of planning could conceivably be used as the basis of a system with a control structure for understanding, just as the theory of planning I will present below could be used by a system with a generative orientation. In fact the theory I present in the next section was used by Meehan (1976) to build a program that generates stories.

The aspect of STRIPS-type plans that precluded their use in my system had more to do with the nature of the domain than the nature of the program. The domain of STRIPS is a simple blocks world. The domain of PAM is that of human interactions. There just isn't that much similarity between them. It would be very strange to hear a story like the following:

(2) John had a box in one room and a box in another room. He wanted to have both of the boxes in the same room, so he went into one room and went over to a box. John bent down and put his hands around the box. Then John stood up holding the box. John brought the box out of the room. Then he carried it into the other room. John bent down and put the box on the floor. Then he moved his hands away from the box and stood up.

Story (2) is an example of a story that could be told about the domain in which STRIPS-type planning is applicable. Clearly, story (2) is peculiar. First of all, it's extremely uninteresting. It's hard to imagine why someone would bother to tell it in the first place. Perhaps STRIPS-type operators could be used to understand this story, but there would be little point in building a story understander that could understand stories no one would ever bother to tell.

The difference between the plans needed for robot planner and the plans needed for a natural language processor has to do with the difference between the physical world and the "human" world. Robot planners like STRIPS use plans that are designed to allow the robot to operate upon its physical environment. It might well be necessary for such a robot to have plans of or even beyond the level of detail of story (2).

In contrast, natural language is a medium for communication between people. The information that people can and do communicate to each other is rarely on the level of such plans. Most people would find it difficult to describe the plans they use for tying their shoelaces, for instance. Not only is describing these details difficult to do; it is an inappropriate level of description. If a person wished to communicate the fact that he had tied his shoelaces, he would not bother spelling out all the steps involved. Stating that he had tied his laces would be sufficient.

Thus robot plans are fit for a domain that requires a different level of detail than is appropriate for natural language processing. The "human" world is not merely grosser than the physical one, however. Some aspects of the world in which people function have no counterpart at all in the physical world. For example, robot planners view their world as a set of inert objects, whereas many of the plans humans use presume the existence of other people. Imagine trying to use robot plans to understand the sentence

(3) John asked Mary to go out to dinner with him.

To understand sentence (3), a reader had to realize that asking is a way

of persuading someone to do something. Persuasion is a concept that presumes the existence of another planner who can be manipulated. Since manipulating another planner is an issue foreign to problem solving and robot planning, the plans from those domains give little help in representing the plan implicit in sentence (3).

There is another way in which a theory of planning for problem solving differs from a theory of planning needed for natural language processing. A problem solver never has to worry about what its goal is or where it came from in the first place; a problem to solve is supplied by the user. However, suppose a natural language understander were told that John wanted to get something to eat. The understander would probably infer that John was hungry, because hunger is a state that can give rise to goals.

A theory of planning that enabled an understander to make this inference must therefore organize knowledge about where goals come from. Since the problem of inferring what the goal is or why it came into existence never arises in problem solving, the theory of planning needed for language processing must include new kinds of planning knowledge that were not needed for the solution of problems.

In sum, the particular set of plans used by problem solvers and robot planners is not the right set of plans for natural language understanding. The plans used in problem solving tend to be of too low a level to be useful in story understanding. These plans are geared for a world in which there is but one planner, and therefore ignore the richness of human planning which presumes the existence of other animate planners. The task of story understanding also raises a number of issues that do not come up in tasks of problem solving. Problem solvers need not determine what problem to solve or why, but a natural language system needs to be able to infer what the goal is and why it arises.

## 3.3 Plans, Goals, Themes and Beyond

To build a story understanding program, a theory of planning is needed that describes the kinds of plans people use, and does so at the right level of detail. The theory of planning I used as a starting point for PAM was based on that of Schank and Abelson (1977). This theory was useful for my purposes because it possessed the following features:

1. It makes a specific commitment to the particular intentional entities that a natural language understanding system needs.

2. It represents these entities on a level of detail that appears to correspond well with the level of detail expressed in linguistic utterances.

3. It is a theory that is primarily concerned with problems of planning in the "human" rather than in the physical world.

The following is a brief overview of Schank and Abelson's system.

### 3.3.1 Schank and Abelson's Theory of Planning

The Schank and Abelson theory of intentionality consists of the following elements:

1. A classification of goals into several categories.

2. A set of primitive planning structures.

3. A classification of themes.

Schank and Abelson classified goals into categories that reflect differences in how each class of goals could be achieved. They distinguished the following classes:

1. Satisfaction goals (S-goals) arise from recurring biological needs. This class includes such goals as Satisfy-hunger, Satisfy-sex, Satisfy-sleep.

2. Enjoyment goals (E-goals) are optionally pursued for pleasure. Enjoy-entertainment and Enjoy-eating are examples of enjoyment goals. Note that the same activities pursued for the fulfillment of an S-goal can also be pursued leisurely for enjoyment.

3. Achievement goals (A-goals) include the establishment of abilities, possessions, and social positions. For example, Achieve-good-job and Achieve-skill are achievement goals.

4. Preservation goals (P-goals) involve the preservation or maintainance of some desireable state. Preserve-health and Preserve-possessions are instances of preservation goals. A special class of P-goals called Crisis goals deal with immediately, rather than potentially, damaging situations.

   The semantics of preservation goals has the following important property. Preservation goals, at least as I have used them, exist only when there is some threat to the state to be preserved. For example, a character does not have the goal P-health unless his health is threatened. This definition of a preservation goal might at first seem counter-intuitive, since it seems to imply that a person does not normally wish to preserve his health, except on certain occasions.

   However, it is even more disturbing to make the alternative assumption that goals such as P-health are always present. Making this assumption would muddle the semantics of the term goal beyond recognition, since a character could then have goals that could not be acted upon, could not be achieved, and yet would fail to be achieved.

   To capture the intuition that there are states like health that are usually are worth preserving, a theme is postulated that gives rise to preservation goals under the appropriate

circumstances. Thus a person always has a theme that may generate preservation goals, but will not have an active preservation goal unless a threat arises. This use of preservation goals is similar to McDermott's notion of policies in problem solving (McDermott, 1977).

5. Instrumental goals (I-goals) are goals that are realized in the pursuit of other goals.

6. Delta goals (D-goals) are actually a special case of I-goals that both occur frequently and can be achieved in a multitude of ways. Delta-goals include the goals of changing one's location (D-Prox), changing one's knowledge state (D-Know), changing control of an object (D-Cont), obtaining social permission to do something (D-Soccont), and employing an agent on one's behalf (D-Agency).

Because of the richness and frequent occurrence of D-goals, Schank and Abelson have centered most of their planning concepts around these entities. That is, most of the other goals classified have fairly inflexible methods of fulfillment: To satisfy hunger one can do little other than eat; to enjoy a play one must attend to it. Also, it is unlikely that satisfying one's hunger or enjoying a play will occur in the service of another goal.

In contrast, there are numerous ways one can obtain possession of an object or find out a fact. For example, an object could be had by bargaining, or asking for it, or threatening its owner, or stealing it, etc. Moreover, D-goals will pop up again and again in the service of other goals. Satisfying hunger, for example, requires the possession of food and being near the food. Thus a great deal of the variability in stories comes from the occurrence of D-goals in the situations described. To understand these stories, then, the reader must have the planning knowledge needed to understand those situations in which D-goals occur.

Schank and Abelson express the planning knowledge of how to achieve D-goals in terms of primitive planning elements called planboxes. A planbox is a unit of planning that is usually not broken down into smaller steps. To use a planbox, a planner must first make sure that all the preconditions of the planbox are met, and then the planner simply does the actions described in the planbox. Thus the structure of Schank and Abelson's planning units are not radically different that those used by problem solvers or robot planners. The important differences are the level at which the plans are specified (they are less detailed) and the domain to which they are applicable (the human world as opposed to the physical world).

For example, ASK is a planbox that can be used for many of the D-goals. The actions in the ASK planbox consist of making a request of someone, and then waiting for a response. In order to use ASK, a planner must insure that a number of preconditions are met. ASK requires that the planner can communicate with the person to be asked, for instance. To insure that this is the case, the planner might have

to achieve another D-goal first, such as D-prox. By chaining together combinations of D-goals and planboxes in this manner, a great number of different plans can be constructed to achieve a particular D-goal, or to achieve another goal in which that D-goal had arisen. Thus planboxes are the building blocks of plans.

Schank and Abelson recognize the following planboxes:

1. A set of planboxes called the Persuade package that are applicable to most of the delta goals:

   1. ASK - Expressing a request to someone.

   2. INVOKE-THEME - A variation of ASK in which the planner reminds the person to be persuaded of an existing thematic relationship between them.

   3. INFORM-REASON - Letting someone know that there exists a reason why they should perform the requested action.

   4. BARGAIN-OBJECT - Offering to give an object in exchange for fulfilling the request.

   5. BARGAIN-FAVOR - Offering to do something that is desirable to the other person in exchange for fulfilling the request.

   6. THREATEN - Promising to do something undesirable if the other person does not agree to cooperate.

2. There are two planboxes specific to D-Cont:

   1. OVERPOWER - Physically incapacitating someone.

   2. STEAL - Taking without the knowledge of the possessor.

3. Two planboxes specific to D-Soccont:

   1. USURP

   2. GO OVERHEAD

4. Four planboxes specific to D-Prox:

   1. RIDE ANIMAL

   2. USE VEHICLE

   3. USE PUBLIC TRANSPORTATION

   4. USE SELF

Schank and Abelson also postulate three kinds of themes, or goal generators. These are the following:

1. Role themes gave rise to goals involving a particular role. For example, being a garbage man may cause one to have the goal of picking up the garbage.

2. Interpersonal themes deal with relationships and attitudes between people. If John loves Mary and she becomes endangered, then John may want to save Mary because of a goal generated by the love theme.

3. Life themes describe a person's general life ambitions. The life theme of being successful could give rise to many goals throughout a person's life.

3.4 Utility of the Model

The point of creating PAM was to build a computer model that could understand situations it had not previously encountered. The Schank and Abelson theory of intentional structures is useful in this regard for the following reasons:

1. It provides a way of organizing knowledge about actions so that connections between events can be inferred that are not "canned" into the program's memory structures.

2. The same organization makes it easy to integrate new knowledge into the system and let it interact with previous knowledge.

For example, consider the following story:

(4) John was hungry. He asked Bill where a restaurant was. Bill said there was one about a mile away. John got into his car.

Before PAM reads this story, the program may never have seen a story in which a person asks about the location of something, and then gets into a car. However, PAM is able to understand this situation because it knows that finding out something is often instrumental to going someplace, and that getting into a car is usually instrumental to using the car to go somewhere. The program is able to relate each input to some knowledge structures for which it has a connection, in this case, to the goals D-Know and D-Prox, and thus relate the two story sentences. Thus these intentional structures provide a convenient level at which to organize knowledge for application to story understanding.

The utility of the Schank and Abelson intentional structures can also be seen by the way in which they enable the system to be extended. For example, after doing story (4) above, I wanted PAM to understand a

version of the Michelin guide story:

(5) Willa was hungry. She picked up the Michelin guide. She got into her car.

This story bears some resemblance to the previous story. From the viewpoint of the story's intentional structure, the story is identical to story (4), except that its planner uses a different plan to find out the location of the restaurant. Thus by supplying PAM with the knowledge of this plan, PAM should be able to understand this story without any modification.

This in fact was what happened. I had to add knowledge about this plan to PAM in the form of rules like the following:

If

a person wants to have a book,

then

that person may want to read that book.

If

1. a person wants to read a book,

2. and the book is a source of information,

then

that person may have the goal of knowing the information contained in the book.

I also had to add the information that the Michelin Guide is a source of information about where to find restaurants. Thus when PAM reads that Willa picked up the guide, it uses the first rule to infer that Willa wanted to read it. Next, to explain why she wanted to read it, PAM uses the second rule to infer that Willa's goal was to find the location of a restaurant. From this point onward, the processing of the story is the same as in the previous case: PAM infers that Willa wanted to go to a restaurant in order to do the restaurant script in order to satisfy her hunger. The next sentence is explained as a plan to go the restaurant, since PAM has already inferred that Willa had the goal of being at one.

Thus by organizing knowledge around the intentional structures of goals, plans, etc., it is a relatively straightforward task to introduce new information into the system and apply it to the understanding process together with previous information. New information can describe what intentional structures might explain an input. Once a path is found to a structure already in the system, all the knowledge about that structure is now also available to the understanding task. In the case of the Michelin guide story, information about the guide and

about reading were added to find a path to the intentional structure D-Know. Once this path was found, PAM's knowledge about D-Know could be used to find the rest of the explanation for that sentence.

### 3.5 A Problem

If all stories conformed to Schank and Abelson's knowledge structures, then supplying PAM with knowledge about these entities would be sufficient for understanding stories involving people's goals. Unfortunately, when I tried to implement this theory, I ran into a number of cases for which the theory was inadequate.

The primary defect with Schank and Abelson's theory was that the stories it could handle did not appear to conform well to the actual stories people read. That is, the Schank and Abelson theory could be used to understand simple stories in which a character has a goal and then takes some steps to achieve it. But these simple sequences did not in themselves constitute reasonable stories.

For example, contrast the following two paragraphs:

(6) John loved Mary. He asked her to marry him. She agreed, and soon after they were wed. They were very happy.

(7) John loved Mary. He asked her to marry him. She agreed, and soon after they were wed. Then one day John met Sue, a new employee in his office, and fell in love with her.

Story (6) is typical of the simple goal-based stories that can be represented in terms of Schank and Abelson's system of plans and goals: A character has a goal (wanting to marry Mary) generated by a theme (being in love with her) and pursues a plan (asking her) that results in the goal being fulfilled. While story (6) is cogent enough, it is a bad story. That is, it is hard to imagine that someone would actually bother telling it, or that anyone would want to read it.

In contrast, story (7) is a much more interesting story. Story (7) is not merely cogent as is story (6). In addition, story (7) expresses a <u>human dramatic situation</u>. A human dramatic situation is a sequence of goal-related events that contains some <u>problem</u> for a character. For example, in story (7), the problem is that John loves someone, but is already married to somebody else.

The notion that problems form the basis of many stories was noted by a number of people, in particular, by Rumelhart (1976). Rumelhart uses the notion of a problem in his theory to refer to any situation involving a goal. My concept of a problem differs from Rumelhart's in that it requires a character to have trouble fulfilling his goal. For example, Rumelhart's theory does not make the distinction between story (6) and (7) above, although one story clearly appears to be more interesting than the other.

I define a story problem as a difficulty in fulfilling a goal. The Schank and Abelson model could be used for understanding stories in which a character fulfills a goal in a straightforward fashion. But stories in which goals are problematic are generally more complicated. Thus to build a program capable of understanding the interesting dramatic situations that occur in real stories, a more powerful theoretical apparatus is needed.

The reason a more powerful theoretical apparatus is required for these stories is that most problematic situations involve more than one goal at a time. The Schank and Abelson model can be used to understand events that are related to a single goal, but not those events whose explanations involve more that one goal. Thus to build a story reader capable of understanding stories that contain problems, it is necessary to develop a system capable of recognizing, describing, and understanding situations involving a number of goals.

For example, consider the following story:

(8) John told Bill he would break his arm if Bill didn't give John his bicycle. Bill got on the bicycle and rode away.

To understand why Bill left, the understander must realize that Bill as well as John had goals that were active in this story. Here Bill is trying to avoid losing his bicycle to John, and is also trying to avoid getting hurt. He is trying to accomplish these goals by running away from John. To understand (8), then, the reader must not only infer what Bill's goals were, but it must recognize the relationship between Bill's goals and John's. These goals are <u>competing</u> with one another. If John fulfills his goal, then Bill will fail to fulfill his, and vice-versa.

To understand this story, the reader must realize the relationship between these characters' goals. But the Schank and Abelson system provides no way of recognizing or representing this relationship.

A similar situation arises when an individual has more than one goal. For example, consider

(9) John wanted to watch the football game but he had a paper due the next day. John decided to watch the football game. John failed Civics.

Understanding the relationship between watching a football game and failing a course is not covered by any of the planboxes in the system presented above. This situation is explainable only if the reader realizes that John's goals are in conflict with one another because of a limitation of time resources. That is, the notion of a <u>goal conflict</u> must be introduced to understand situations in which a character has more than one goal at a time.

Another planning notion required for story understanding involves planning for a number of goals at once. For example, consider the story

(10) John was tired of frequenting the local singles' bars. He decided to get married.

From within the Schank and Abelson framework it is difficult to explain why John decided to get married. To understand how getting married could replace going to a singles' bar, the reader must recognize that John had a goal that was arising repeatedly. Instead of planning for this goal each time it arose, John decided to make it easier to achieve this goal by getting married to someone. The fact that a state can subsume goals that arise repeatedly is an aspect of planning not covered by the Schank and Abelson system.

In sum, Schank and Abelson's approach is useful for understanding a broad range of stories involving goals, but it is not capable of dealing with the more complicated goal situations that arise in real stories. These situations include the cases where there are a number of characters with opposing goals, where an individual has goals that are in conflict, and where the planner is dealing with more than one goal at the same time, as in the case of recurring goals.

The rest of this thesis is primarily concerned with these more complicated goal situations. To build a story understanding system, we need to know what these situations are, how they can be represented, and how they can be recognized. We also need to know what kind of knowledge is required to perform these tasks.

In the following chapters I examine these questions for several complicated goal relationships. The situations in which these relationships appear constitute a significant segment of the stories about human dramatic situations. That is, I found that these situations were prevalent in many actual stories. The goal relationships that I found to dominate stories are the following:

1. Goal Conflict

   A goal conflict is a situation in which one character has several goals such that the fulfillment of one goal will preclude the fulfillment of the others. For example, story (9) above is an instance of a goal conflict because John's goal of watching the football game may interfere with his other goal of writing his paper. Goal conflicts are treated in chapters 6 and 7.

2. Goal Competition

   Goal competition refers to those situations in which several characters' goals may interfere with one another. For example, story (8) above contains an instance of goal competition. John's goal of possessing Bill's bicycle cannot be fulfilled along with Bill's goal of preserving possession of the bicycle. If Bill succeeded in preserving possession of the bicycle, then John would have failed to fulfill his goal. Goal competition is discussed in chapters 8 and 9.

3. Goal Concord

Sometimes several characters in a story will perform actions amenable to each other's goals. For example, consider the following story:

(11) John and Bill hated Fred but he was bigger than either of them. They decided to gang up on him.

Here John and Bill make an alliance to overcome a mutual foe. John and Bill each have the goal of hurting Fred, and therefore join forces in a situation in which neither of them could accomplish this goal alone. Goal concord is dealt with in Chapter 10.

4. Goal Subsumption

Goal subsumption refers to a situation in which a character's plan is to achieve a state that will make it easier for a character to fulfill a recurring goal. For example, story (10) above contains an instance of goal subsumption. In this story, John decides to get married in order to make it easier to achieve the goals he had been achieving previously by going to a singles' bar. Goal subsumption is described in Chapter 4 and 5.

These particular goal relationships were chosen because the situations to which they give rise account for a large class of story problems. That is, dramatic situations involve a difficulty in fulfilling a goal, and these difficulties often arise due to goal interrelations. In particular, goal relationships can give rise to these problems:

1. Goal Conflict - If a character is unable to resolve a goal conflict, then one of that character's goals may fail. Thus goal failure due to goal conflict, and attempts to resolve goal conflict both provide interesting story situations.

2. Goal Competition - As with goal conflict, the existence of competitive goals implies that some character may have trouble fulfilling his goal. Interesting stories therefore exist involving goal failure due to goal competition, struggles against the plans of other characters, and attempts at easing the competition.

3. Goal Subsumption - Goal subsumption gives rise to dramatic situations when a subsumption state is terminated. For example, if John is happily married to Mary, and then Mary leaves him, all the goals subsumed by their relationship may now be problematic - John may become lonely, and miss his social interactions with Mary, for instance. Closely related to problems based on goal subsumption are those caused by the elimination of normal physical states. For example, becoming very depressed or losing a bodily function can give rise to the

inability to fulfill recurring goals, and can therefore generate some interesting problems.

4. Goal Concord - Goal concord does not give rise to problems as much as it is involved in their solutions. For example, two people with concordant goals may be unable to fulfill their goals individually, but may be able to work together to achieve them.

Goal difficulties can also be created in the case of a single goal which is inherently difficult. The problem of determining the difficulty of a goal is discussed in Chapter 12.

# CHAPTER 4

# GOAL SUBSUMPTION

Characters often plan ahead. If a character has a goal that occurs repeatedly, he can try to establish a state that makes it easier to fulfill that goal each time it arises. A state that makes it easier to fulfill a recurring goal is called a goal subsumption state. Important goal subsumption states include owning a functional object, having access to a continuous supply of a resource, being in a social relationship to someone, and knowing a useful fact. For example, owning a functional object makes it easier to fulfill those goals to which the object's function applies.

A reader needs knowledge about goal subsumption to understand a number of story situations. For example, a character who has difficulty fulfilling a recurring goal may want to establish a state that subsumes this goal. Since this goal arose out ot the need to subsume a recurring goal, goal subsumption constitutes an explanation for this goal's presence.

This chapter describes goal subsumption and gives several important instances of its use. The next chapter describes how to process three kinds of story situations in which goal subsumption occurs: Situations in which a goal subsumption state is established, situations in which an existing goal subsumption state is replaced by a new subsumption state, and situations in which a goal subsumption state is terminated.

## 4.1 Introduction

Consider the following stories:

(1) John had to go to Hartford to pick up a friend at the airport. He asked Fred if he would lend him his car.

(2) John got a job in Hartford. He asked Fred if he would sell him his car.

Story (1) is typical of the kinds of goal-based stories discussed in previous chapters: A character has a goal and pursues a plan directed at fulfilling this goal. In (1), John had the goal of being in Hartford, and tried to obtain control of a vehicle in order to make the trip.

Story (2) at first appears to be similar to story (1). The character in this story also acts on a goal. In fact, the goal seems to be the same as the goal that occurs in story (1): John needed to be in Hartford in story (2), just as he did in story (1).

However, the plans chosen to fulfill these goals are quite different. Instead of borrowing a car as in story (1), in story (2) John decided to purchase one. Moreover, the plan chosen in one story is inappropiate in the other. Consider

(3) John had to go to Hartford to pick up a friend at the airport. He asked Fred if he would sell him his car.

This story seems strange because the plan chosen does not seem to correspond well to the needs to the planner. Taking one trip to the airport does not seem to merit purchasing a car.

Moreover, the inferences an understander needs to make after determining the goals in each story are drastically different. For example, in story (1), the reader expects that John will drive to Hartford shortly after getting Fred's car. Thus if (1) were extended to

(4) John had to go to Hartford to pick up a friend at the airport. He asked Fred if he would lend him his car. Fred gave him the car and John got inside.

the reader would infer that John was heading off to the airport. However, this inference would not be made in story (2). If (2) were extended to

(5) John got a job in Hartford. He asked Fred if he would sell him his car. Fred sold him the car and John got inside.

probably most readers would conclude that John was going to drive home. Even though the reader knows John bought the car to commute to Hartford, he would not assume John was going to go there immediately.

Thus the goals in stories (1) and (2) have some important differences. The plan used to fulfill the goal in one story is not applicable to the goal in another, and different inferences are made from each goal. Thus the goal in story (2) must be something other than to be in Hartford. The problem is to determine what kind of goal it is, what sort of plans are applicable to it, and what its relationship is to the goal present in story (1).

#### 4.1.1 Goal Subsumption

The difference between story (2) and story (1) is that story (2) involves the recurrence of a goal. In (1) John had the goal of being in Hartford and borrowed a car in order to drive there. By contrast, in (2) John had a job in Hartford, which meant that he would have to be in Hartford each day. John is continually confronted with a recurring goal, in this case the goal of being in Hartford. The story is not concerned with John attempting to go to Hartford, but with John dealing with the problem of having to commute to Hartford daily.

To understand story (2), a reader must realize that John is using a plan that is directed not at a single goal, but at a multitude of potential goals. The goal in story (2) is not to be in Hartford, but to make it easier to achieve this goal each time it arises. Owning a car makes it easier to deal with being in Hartford each morning because a means of transportation is then always available. A person who owns a vehicle need not worry about obtaining control of one, as was necessary in (1). In other words, owning a car <u>subsumes</u> John's recurring goal of being in Hartford every day.

Doing something that makes it easier to fulfill a recurring goal is called <u>goal subsumption</u>. Goal subsumption is a very general planning strategy that allows a planner to pursue many goals at once. Goal subsumption is possible whenever a character can anticipate having a number of goals, and there exists a set of plans for these goals all of which share a common precondition. If the character can then fulfill this precondition so that it holds each time he pursues one of these goals, then he will be saved the work of fulfilling that precondition as each individual goal arises.

A story understanding system must be able to recognize that a situation involves goal subsumption in order to explain the behavior of a character acting in that situation. For example, consider the following story:

> (6) John was feeling lonely every evening. He decided to get married.

Here John has the recurring goal of alleviating loneliness. Being married to someone subsumes this goal by providing John with a companion. To alleviate loneliness, a person can use a plan that entails social interactions with other people. Social interaction has the precondition that a person be available with whom one can interact. Marrying someone assures that the person married will be available for social interactions over a fairly long period of time. Thus an understander could explain John's behavior in story (6) only if it inferred that the state of being married to someone fulfills the "have companion" precondition of the "social interaction" plan for the recurring goal of alleviating loneliness.

A goal subsumption relationship contains the following components:

1. A set of goals that arise repeatedly. In story (6), this was the set of recurring Enjoy-company goals.

2. A set of plans for these goals, all of which share a common precondition. In story (6) the plan of SOCIAL-INTERACTION was applicable to each goal.

3. A state that meets the common precondition and which endures over a time span that encompasses repeated instances of the set of goals. Above, being married to someone fulfills the "have willing companion" precondition of the SOCIAL-INTERACTION plan by providing a person who is obliged to interact with the planner.

Knowledge about goal subsumption is important in story understanding because it allows the reader to infer a reason behind a character's goal. For example, in (6), a reader would need to infer that the reason John wanted to get married was because he thought it would help alleviate his recurring loneliness problem. To make this inference, the reader needed to know that having a goal repeatedly might cause a character to want to establish a state that can subsume that goal. That is, the reason John wanted to get married in (6) can be described in terms of his desire to make it easier for him to enjoy companionship.

### 4.2 Types of Subsumption States

To understand how achieving a state can subsume a goal, the reader must know what the existence of that state entails. In the above example, for instance, the fact that marriage imposed obligations upon its participants is needed to understand how marriage could subsume loneliness. This section gives some additional examples of common subsumption states, and outlines the knowledge required to understand them.

#### 4.2.1 Ownership of A Functional Object

Ownership of a functional object is a subsumption state. Functional objects are objects that are normally used for particular plans. For example, a car is a functional object that is normally used for transportation; money is a functional object that is normally used in trading. Functional objects are discussed in detail in Chapter 11.

The state of owning an object that has a functional use can subsume the goal for which that object is used. For example, in

(7) John decided to become a paid assassin. He bought a gun.

Assassin's frequently have to kill people. In order to make it easier

to fulfill this goal each time it arises, John decided to subsume the recurring goal by acquiring ownership of a weapon. Once this state is realized, John can use the OVERPOWER planbox to murder his victims without having to deal with the "have weapon" precondition for this planbox. Owning an object assures the owner the right of access to the object.

The goal subsumption relationship in story (7) consists of the following components:

Recurring goal:
 someone IS HEALTH(-10)
  Planner = JOHN
  Victim = someone

Subsumption state: GUN IS OWNED(JOHN)

Plan: OVERPOWER
 Planner = JOHN
 Weapon = GUN
 Victim = someone

Constraints: GUN IS CONTROLLED(JOHN)

Stored in memory under the concept of ownership is the fact that owning an object subsumes those goals for which the object is used. Story (7) is an example of how ownership of a non-consumable object can subsume a recurring goal. Non-consumable objects are those functional objects that are not used up when they are used in a plan. Ownership of non-consumable objects can therefore serve to subsume a large number of recurrences of a goal. In contrast, consumable objects are functional objects that are used up in some way. For example, food is a consumable object that is used up when it is eaten (The notion of consumable objects is defined and discussed in Chapter 11).

Ownership of a consumable object can subsume as many goals as the object will provide for. Consider the story

(8) John decided to go camping for a week. He went to the supermarket and bought some groceries.

A reader must infer that John bought enough groceries for a week's worth of meals. To make this inference, the reader must know that the usual plan for fulfilling hunger, i.e., eating, requires its user to control some food, and furthermore, that the food will be consumed in the course of carrying out the plan. The reader must also know that the goal of satisfying hunger will arise many times during a week. That is, the goal of satisfying hunger is generated by the hunger drive, and this drive recurs with a certain periodicity. Therefore, for John to subsume these goals for a week, John must include in his purchase some groceries for each instance of an anticipated Satisfy-hunger goal. John was fulfilling the precondition of having some food for each meal by purchasing the food all at once.

For example, suppose a reader of story (8) were asked the question

Q1) How much did John buy?

An appropriate answer would be a week's worth of groceries. In contrast, suppose the reader were told the following story:

(9) John was feeling hungry. He went to the supermarket and bought some groceries.

Now if the reader were asked question Q1, an appropriate response would be that John bought enough food for a meal. The difference between these two answers is based on the number of goals in each story for which going to the grocery is instrumental. In story (8) John was subsuming a number of goals by buying food, while in story (9), buying food is instrumental to a single goal. Here the notion of goal subsumption is used to determine the number of goals that the precondition is instrumental to, and thus compute the amount of groceries John purchased.

### 4.2.2 Tapping a Stream

A stream is a continual source of a consumable substance. For example, a river would be an instance of a physical stream supplying water. A job is an example of a social stream that supplies the employee with money. Streams are discussed in a note at the end of this chapter.

If a planner can establish some means of tapping a stream of some consumable, then he can subsume those goals for which the consumable may be used. For example, consider

(10) John was hungry because he lost his job.

An understander needs to know that having a job subsumes goals involving money by creating a continual supply of money. This knowledge is needed to infer that the reason John was hungry was that he could no longer afford to buy food after his stream of money dried up. Thus (10) involves an instance of goal subsumption with the following attributes:

Recurring goal:
DCONT
Planner = JOHN
Desired-Object = some object

Subsumption state: JOHN HAS JOB

Plan:
BARGAIN-OBJECT
Planner = JOHN
Give-Object = MONEY
Desired-Object = some object
Constraints: MONEY IS CONT(JOHN)

That is, money can be used to buy things. Buying is an instance of the plan of bargaining for something, and bargaining can be used to achieve goals involving change of control of an object. Having a job can therefore subsume goals involving change of control of an object by providing a stream of money with which the desired objects can be bought.

### 4.2.3 Being in a Social Relationship

Being in a social relationship imposes a set of obligations upon a person. An obligation is a belief held by a person's society that a person must act in a particular way in certain social situations. For example, being married implies that each person in the marriage relationship is obligated to help fulfill the other's emotional and sexual needs.

Since social relationships tend to be relatively long term, the obligations they impose upon another person may serve to subsume a planner's goal. For example, consider the story

(11) Mary was tired of working for a living. She decided to quit her job and get married.

To understand the relationship between the sentences of (11), the reader must know that in a traditional marriage relationship, the woman need not work because her husband is supposed to look after her financial needs. In (11) Mary seems to feel that if she were married to someone, then that person would be obligated to support her. This would subsume the same economic goals as having a job, namely, those goals for which money is a prerequisite.

The concept of "traditional marriage" is an instance of knowledge about the way social relationships exist in the minds of members of a particular cultural group. That is, the reader needs to know how people view the obligations imposed on them by their society, even if the reader does not necessarily adhere to that view. For example, a given understander may believe in "full partnership marriage", "open marriage", "Mormon marriage", etc. Nevertheless, that reader would need

to understand the concept of a traditional marriage if he was to understand the motivations of a person who was a member of a culture in which this concept of marriage functioned.

Since there is typically more than one obligation imposed upon the member of a relationship, social relationships usually subsume more than one goal. For example, being married to someone usually subsumes a number of recurring goals in addition to those for which money is instrumental. Since being married obligates each partner to have sex with the other, and since having a willing partner is a precondition for having sex, then marriage subsumes the recurring Satisfy-sex goal. Also, since marriage requires the partners to live together, then it subsumes the recurring Enjoy-company goal to which being near a loved one is instrumental. That is, the definition of a traditional marriage includes the following information:

Traditional Marriage

Obligations:

Spouse1 achieve Satisfy-sex(Spouse2)

Spouse1 achieve Enjoy-company(Spouse2)

Spouse1 achieve D-cont(Spouse2,possessions)

That is, a traditional marriage obligates one spouse to satisfy the sexual, emotional and material needs of the other. Of course, this description of marriage is *not meant to be complete*. There are many other obligations involved in marriage, such as obligations to children, etc., and many other aspects of this relationship that are not relevant to its function as a goal subsumption mechanism.

### 4.2.4 Knowing Something

Another class of goal subsumption relationships has to do with committing a fact to memory. Knowing a fact is often instrumental to many goals, so if these goals recur, having that fact available will circumvent the need to find it out. For example, consider

(12) When John became treasurer of the club, he memorized the combination to the safe.

To understand why John memorized the combination, the understander must realize that the goal of opening the safe is liable to recur for John. This inference is based on the fact that John now has the "treasurer" role theme, and this theme may give rise to goals involving the use of the safe. The reader must also understand that knowing the combination to the safe would make it easier to open the safe each time the need arose. Thus (12) involves a subsumption state with the components:

Recurring goal:
OPEN
Planner = JOHN
Object = SAFE

Subsumption state: combination IS MLOC ( LTM of JOHN )

Plan:
$SAFE-OPENING SCRIPT
Planner = JOHN
Safe = SAFE
Constraints: combination IS MLOC ( STM of JOHN )

To understand story (12), a reader needs to perform the following processes: First, upon hearing that John has become treasurer of his club, the reader must recognize that John now has a new role theme. This social role gives rise to the prediction that John will now have recurring goals generated by this role. When the reader reads the second sentence and tries to find an explanation for it, the reader must determine what purpose there may be behind memorizing a combination. Memorizing something is a plan for knowing something that is likely to be used frequently. Knowing the combination to a safe is useful for opening a safe, so the reader infers that John anticipates having the goal of opening the safe.

Opening a safe is instrumental to using a safe, which is in turn used to store or retrieve money. The social role of treasurer entails having goals involving the transfer of money. Since this role recurrently generates this goal, the prediction set up after reading the first sentence fires off. This prediction states that John's new role theme may give rise to a recurring goal, so the inference is made that John wanted to know the combination to the safe because he anticipated having the goal of opening the safe on numerous occasions.

## 4.3 Understanding Goal Subsumption Stories

There are three kinds of stories in which the notion of goal subsumption is important. They are as follows:

1. Goal Subsumption State Establishment

The simplest situation involving goal subsumption occurs when a character decides to subsume a recurring goal. The character then proceeds to select and execute a plan for this goal. An example of this situation is the following story:

(6) John needed more money to pay his rent. He decided to take another job.

John has the goal of establishing a subsumption state to make

it easier to pay the rent each month, and decides to achieve it by getting a job and establishing a stream of money.

2. Goal Subsumption State Replacement

A variation on Subsumption State Establishment occurs when the character replaces an old subsumption state by a new one. For example, consider

(13) The carpenter's old hacksaw wasn't doing the job any more. He decided to buy an electric saw.

In (13), the carpenter is purchasing an electric saw to subsume his recurring goal of cutting wood. However, owning a hacksaw previously subsumed this goal. Thus the new state replaces an old one in addition by subsuming the same goal by a different plan. Goal Subsumption State Replacement is different that the establishment of a subsumption state because it makes the old state obsolete. For example, the carpenter may sell his hacksaw after (13) because he no longer needs it to do his work.

3. Goal Subsumption State Termination

A subsumption state may be ended, by either chance or volition, and leave a character with a problematic recurring goal. An example of Goal Subsumption termination occurs in the following story:

(14) John and Mary were happy children. During the war, their parents were killed, and they both become street urchins.

Being a child in a family normally subsumes a number of goals for the children. When such a state terminates, then, as it did in (14), the reader can infer that the recurring goals the state subsumed now must be dealt with individually. For example, in (14) the reader might infer that the children didn't get enough attention and had trouble getting enough to eat. In addition, the reader might infer that the children wanted to subsume these goals again, and that they would be eager to be adopted by new loving foster parents.

Each of these particular situations has its own set of inferences and knowledge needed to understand it. The next chapter explores each situation and presents the knowledge and procedures by which a reader could interpret it.

## 4.4 Note on Streams

This section contains a brief description on streams that supplements the discussion above.

A stream is a continual source of a consumable substance. If a planner can establish some means of tapping a stream of some consumable, then he can subsume those goals for which the consumable may be used. For example, consider the following story:

(15) John used to be rich but then he lost all his money. John decided to find a job.

To find the connection between the sentences of (15), a reader must first realize that being rich subsumes many important goals, and that terminating this state ends a subssumption state. The next sentence can then be interpreted as a plan to re-subsume these goals by achieving a different subsumption state.

Having a job appears to be a way of subsuming the same goals as are subsumed by being rich, but by some entirely different method than ownership: Through ownership, a planner can subsume goals by possessing a quantity of a substance; by having a job, the planner is provided with a steady source of the substance. Thus having a job is a way of accessing a continuous stream of a consumable.

The subsumption states involving streams can be described with the following rule:

Rule Stream:

If

a character has access to a stream of a consumable substance,

then

the stream can subsume a number of goals that is a function of the amount of the substance available to the character from the stream, and the amount of substance required for the achievement of each goal.

I have found it necessary to define two types of streams:

1. Physical Streams. These are actual physical objects that "issue forth" a substance. For example, streams of water and wells would be exemplary instances of physical streams. A goal subsumption relationship involving a physical stream implies that the planner have some kind of access to the physical object. As an example, consider

(16) To insure a steady supply of water, John dug a well on his farm.

In (16), John establishes a subsumption state by creating a physical

stream which is then his to use as he likes. To understand (16), the reader would be required to have in his memory the following information about the object "well":

```
WELL
  FUNCTION:
                                                     |--> bucket
   PLANNER <=> PTRANS <--O-- water ---|
                                                     |--< WELL
   CONSEQUENCES TO OBJECT
          - none
```

That is, a well is an object whose function is to supply water. An object is therefore a physical stream if it (i) has a function transfers some substance to the planner, and (ii) there are no consequences to the object. In effect, this may be considered as another way in which ownership of a physical object can subsume a goal.

2. Social Streams. These are reasonably long-term social states that assure an person of a continual supply of a consumable. For example, owning one's own business is an instance of a social stream in which the owner repeatedly performs some action in exchange for money. The interest earned on money in the bank is a social stream were the planner need not do anything. Having a job is a contractual relationships between employer and employee in which a consumable substance (i.e., money) is repeatedly given to the employer in exchange for services rendered. In the example

(17) John was hungry because he lost his job.

an understander needs know that having a job subsumes goals involving money by creating a continual supply of money. This knowledge is needed to infer that the reason John was hungry was that he could no longer afford to buy food after his stream of money dried up. Understanding (17) therefore requires the following representation for "having a job" in the reader's memory:

```
OCCUPATION(one)
  ACTIVITY: ...
  CONSEQUENCES: money IS POSS(one)
  PERIODICITY: ...
```

That is, that someone has a job means that they perform some activity with some periodicity (e.g., five days a week) and get paid money for it. In general, a social situation that causes someone to transfer a resource to someone else at regular intervals constitutes a social stream.

Social streams are often very complex relationships involving other goals For example, having a job may subsume other goals besides having money, such as Prevent-boredom and Enjoy-company, if the job involves interesting activities and interaction with fellow-workers. This knowledge might be needed if a reader were to understand a story like

(18) After John lost his job, he began to get bored.

Here the relationship between losing a job and becoming bored must be inferred. To make this inference, the reader must know that a job can subsume someone's Prevent-boredom goal, and therefore that terminating that subsumption state could result in that person becoming bored.

### 4.5 Summary

Goal subsumption is a way to plan for recurring goals. A goal can be subsumed by establishing a state that fulfills a precondition for a plan for each recurrence of a goal. Some important goal subsumption states include owning functional objects, having access to continuous supplies of resources, being in social relationships, and knowing useful facts.

Goal subsumption occurs in a number of story situations. These include situations in which a goal subsumption state is established, situations in which an existing goal subsumption state is replaced by a new subsumption state, and situations in which a goal subsumption state is terminated.

# CHAPTER 5

# GOAL SUBSUMPTION SITUATIONS

The previous chapter introduced the notion of goal subsumption as a method of planning for recurring goals. A goal can be subsumed by establishing a state that fulfills a precondition for a plan for each recurrence of a goal. Some examples of frequently encountered subsumption states are owning a functional object, having access to a stream, being in a social relationship, and knowing a useful fact.

Knowledge about goal subsumption is needed to understand three kinds of story situations: Situations in which a goal subsumption state is established, situations in which an existing goal subsumption state is replaced by a new subsumption state, and situations in which a goal subsumption state is terminated.

Each type of goal subsumption situation uses knowledge about goal subsumption differently, and thus requires a particular set of inference rules. For example, in a situation in which a character is trying to establish a subsumption state, a reader may be required to infer that the character has a goal of establishing a subsumption state, and that the character's actions constitute a plan to subsume a goal. In a situation in which a subsumption state is terminated, the reader must be able to infer that the character will have to use some other plans to fulfill the goals that were previously subsumed, and that the character may try to establish a new subsumption state.

For example, the following is a goal subsumption termination situation:

> One day, John found that his pocket calculator was missing. He took a pad and pencil out of his drawer.

Owning a pocket calculator subsumes goals involving arithmetic computations. Since the calculator is no longer

AD-A062 629 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 5/7

UNDERSTANDING GOAL-BASED STORIES.(U)

SEP 78 R WILENSKY N00014-75-C-1111

UNCLASSIFIED RR-140 NL

2 OF 4

ADA

062629

available, John must fulfill these goals by other means. In this case, the reader uses this knowledge to infer that John needs to do arithmetic, and that he plans to use the paper and pencil for this purpose.

5.1 Introduction

Goal subsumption plays a role in a number of story situations. In the previous chapter, the following types of goal subsumption situations were identified:

1. Goal Subsumption State Establishment
2. Goal Subsumption Replacement
3. Goal Subsumption Termination

Each of these situations imposes different requirements on the knowledge and processing capabilities of the reader. For example, consider the following story:

(1) The neighbor's dog was keeping John awake each night. He went to the store and bought a pair of earplugs.

Owning a pair of earplugs will enable John to use them whenever the neighbor's dog is preventing him from getting some sleep. To understand why John wants to own earplugs, the reader needs to infer that the goal of owning earplugs arose from the need to subsume a recurring goal. Thus in this story, goal subsumption acts as a goal generator in producing an ownership goal.

In contrast, consider the following story:

(2) The neighbor's dog used to keep John awake at night until he bought some earplugs. Then one day, John misplaced his earplugs. All week long, John was irritable.

An inference that is crucial for understanding story (2) is that John had trouble sleeping all week. To make this inference, the reader needs to realize that owning earplugs subsumed a recurring goal for John, and that when they were lost, John might have trouble fulfilling each goal as it arose. Since the last sentence can be a result of failure to fulfill these goals, the reader infers that John's losing the earplugs led to John's losing some sleep. In this case, knowledge about goal subsumption is used to infer the failure to fulfill a goal.

This chapter explores each goal subsumption situation and presents the knowledge and procedures that a reader could use to interpret it.

## 5.2 Goal Subsumption State Establishment

The desire to subsume a recurring goal occurs in a great number of stories. For example, consider the following:

(3) John loved drinking hot coffee. He bought a thermos.

A reader of (3) must infer that John bought a thermos so he could have hot coffee conveniently. To make this inference, a reader must first realize that enjoying something a lot may periodically give rise to enjoyment goals. Then the reader must find the connection between these recurring enjoyment goals and buying a thermos. Making this connection requires the reader to know that the function of a thermos is to serve as a container in order to preserve the temperature of a liquid, and that owning an object subsumes goals that the object can be used to fulfill. By applying this knowledge to the second sentence of story (3), the reader infers that John bought the thermos to fulfill a recurring goal involving preserving the temperature of some liquid.

Now the reader must explain why John is likely to have this recurring goal. One way to explain the existence of a goal is by finding another goal to which it is instrumental. The reader previously inferred that John has a recurring goal of enjoying hot coffee, so he checks to see if preserving the temperature of a liquid can be instrumental to this goal. Since enjoying hot coffee requires the coffee to be at a certain temperature, the reader infers that John planned to use the thermos to preserve the temperature of the coffee. Thus the reader explains John's buying the thermos as a means to subsume his recurring "enjoy hot coffee" goal.

Story (3) is an instance of a Goal Subsumption State Establishment situation. This situation is comprised of the following components:

1. Goal - The goal of achieving a subsumption state for some recurring goal.

2. Source - Something that causes a character to want to achieve a subsumption state.

3. Plan - A plan (or sequence of plans) for this goal.

4. Action - Actions taken to execute this plan.

Goal Subsumption State Establishment is really an ordinary planning situation in which the principle of goal subsumption causes a planner to want to achieve a particular state. For example, story (3) above would have the memory representation shown in Figure 3.

Figure 3 displays the following information: Something gives rise to a goal, which causes the planner to choose a plan, and then some action is taken to execute that plan. In addition, Figure 3 contains a complicated Source structure. Recall that a source is something that gives rise to goal. Previously, two kinds of sources were encountered: A goal could have arisen from a theme, or it could be instrumental to another goal.

Story (3) requires a third kind of source, called the <u>goal subsumption source</u>. A goal has a goal subsumption source when the reason that goal came into existence was to achieve a subsumption state for a recurring goal. A goal subsumption source has the following components:

Goal Subsumption Source

1. Recurring-Goal = The goal the planner intended to subsume

2. Source-of-Goal = A description of why the recurring goal arises

3. Plan = The plan by which the recurring goal is to be assumed

4. Precondition = The precondition of the plan that the subsumption state fulfills.

The Recurring-Goal component states the immediate goal that the character is subsuming. In Figure 3, this is John's goal of preserving the coffee's temperature. The Source-of-Goal component denotes that the Recurring-Goal arises because it is instrumental to John's goal of enjoying coffee, which arises in turn because John happened to be fond of coffee. In general, the Source-of-Goal component contains either a plan to which the Recurring-Goal is instrumental, or a theme which gives rise to the goal directly.

The Plan of the Goal Subsumption Source is the intended method of fulfilling the Recurring-Goal each time it arises. In Figure 3, the Plan is to contain the coffee in a thermos. Lastly, the Precondition of the Plan that the subsumption state is intended to fulfill is described. In the example above, the precondition is that John own the thermos.

Once a reader has built the representation in Figure 3 from story (3), he could use it to answer questions about the story such as the following:

Q) Why did John buy a thermos?
A) So he could enjoy drinking hot coffee whenever he wanted to.

Figure 3

**Representation of Story (3): John loved drinking hot coffee. He bought a thermos.**

---

```
Source

 ________________________________________________________________
| Goal Subsumption                                               |
|   Recurring-Goal = PRESERVE-TEMPERATURE                        |
|                      Planner   = JOHN                          |
|                      Substance = COFFEE                        |
|   Source-of-Goal = Instrumental-to                             |
|                      Plan    = EAT                             |
|                                 Planner = JOHN                 |
|                                 Food    = COFFEE               |
|                      Goal    = ENJOY-EATING                    |
|                                 Planner = JOHN                 |
|                                 Food    = COFFEE               |
|                      Source = FONDNESS THEME                   |
|   Plan           = USE-CONTAINER                               |
|                      Planner                = JOHN             |
|                      Container              = THERMOS          |
|                      Contained-substance    = COFFEE           |
|   Precondition   = THERMOS IS CONT( JOHN )                     |
|________________________________________________________________|
                          |
                 Goal     |
             ____________________________
            |                            |
            | THERMOS IS OWNED( JOHN )   |
            |____________________________|
                          |
                 Plan     |
               ______________________________
              |                              |
              |  BARGAIN-OBJECT              |
              |   Planner        = JOHN      |
              |   Persuadee      = someone   |
              |   Give-Object    = MONEY     |
              |   Desired-object = THERMOS   |
              |______________________________|
                          |
                 Action   |
        _______________________________________________________
       |                                   |-> PERSUADEE       |
       |   JOHN <=> ATRANS <--O-- MONEY --|                    |
       |             /\ ||                 |-< JOHN            |
       |             || || RESULT                              |
       |             || \/                   |-> JOHN          |
       | PERSUADEE <=> ATRANS <--O-- THERMOS --|               |
       |                                     |-< PERSUADEE     |
       |_______________________________________________________|
```

---

"Why" questions suggest that the reader look at the source of the goal dominating the action mentioned in the query. If the source is Goal Subsumption, the reader can answer that the planner did the action in question because it enabled the him to fulfill the Recurring-Goal (or a goal to which it was instrumental) whenever the planner wished.

To produce the representation in Figure 3, an understander needs to have the following rule:

Rule GS1:

If

a character has a positive attitude toward performing an action,

then

that character may want to establish a state that subsumes the goal dominating that action.

Some other examples where Rule GS1 is useful are the following:

(4) John enjoyed swimming. He put in a pool in his backyard.

(5) John liked to go to the Giants' games. He bought a season ticket.

John has a positive attitude toward swimming in story (4), so Rule GS1 implies that he might want to subsume this enjoyment. Building a swimming pool establishes a subsumption state because it makes it easier for John to swim when he so desires. Being at the Giants' games is fun for John in story (5), so Rule GS1 states that John may want to subsume this goal. Since a season ticket gets one into all the games, owning one fulfills a precondition for a recurring goal.

The process of applying this rule is described for story (3): Liking something is a positive attitude, and drinking coffee is clearly an action, so Rule GS1 can be invoked. Invoking this rule during understanding means that an expectation is created looking for the subsumption of the goal dominating the action. The goal that dominates eating is either Enjoy-Eating or Satisfy-Hunger. Since the reader was explicitly told that John liked drinking coffee, he can infer that it is the Enjoy-Eating goal that the expectation should look to subsume.

Recall that expectations are allowed to look at both incoming events and any bottom-up inferences generated from these events. The expectation just created looks for a goal of attaining a subsumption state for Enjoy-eating.

When the understander encounters John buying a thermos, it starts making bottom-up inferences from this event. One bottom-up inference made about buying is that John has the goal of owning the thermos. Whenever a bottom-up inference is made, each expectations checks to see

if the inference meets its condition. Thus the subsumption expectation examines this inference to determine if it could be an attempt at subsuming John's Enjoy-Eating goal. The expectation uses the subsumption state recognition algorithm diagrammed in Figure 4.

This algorithm is applied to the inferred goal of John wanting to own a thermos with the following results: Step 1 of the algorithm asks if John's owning a thermos would normally subsume any goals. Ownership of an object that has a function subsumes those goals for which the object may be used (This is a rule about objects described in Chapter 11). Thus to compute the goals that ownership of a thermos subsumes, the function of a thermos must be examined.

Figure 4
Subsumption State Recognition Algorithm

```
STEP 1:
          ______________________________
         |                              |
         | Does the state at which      |  no
         | the goal is aimed normally   | =====> The expectation
         | subsume any goals?           |        has not been met.
         |______________________________|

                      ||
                      || yes
STEP 2:               \/
          ______________________________
         |                              |
         | Are any of these goals       |  yes
         | the same as the goal the     | =====> The expectation
         | planner is trying to         |        has been confirmed.
         | subsume?                     |
         |______________________________|

                      ||
                      || no
STEP 3:               \/
          ______________________________
         |                              |
         | Could one of these goals     |  yes   The expectation
         | be instrumental to the goal  | =====> has been confirmed.
         | the planner is trying to     |
         | subsume?                     |
         |______________________________|

                      ||
                      || no
                      \/
        The expectation has not been met.
```

The function of a thermos is to contain liquids in order to preserve their temperature. Since using a thermos is a plan for preserving the temperature of a liquid, owning a thermos is a way to subsume goals of preserving the temperature of liquids. Thus Step 1 of the algorithm computes that Preserve(John, temperature of liquid) is a goal that John may want to subsume.

The algorithm then compares this goal with the goal the expectation predicted John would try to subsume. In this case, the expectation predicted that John would try to subsume Enjoy-eating(John, hot coffee). Since Enjoy-eating(John, hot coffee) is not the same as Preserve(John, temperature of liquid), the algorithm goes on to Step 3.

Step 3 checks to see if the goal inferred bottom-up could be instrumental to the goal the planner is aiming to subsume. That is, could PRESERVE(John,temperature of liquid) be instrumental to EAT(John,hot coffee). Since the plan for Enjoy-Eating is normally EAT, the algorithm checks PRESERVE(John,temperature of liquid) against the preconditions of EAT(John,hot coffee). Eating something while it is is a particular state requires the object to be in that state, so preserving such a state is instrumental to eating that object. Thus the algorithm infers that owning a thermos subsumes a goal instrumental to John's recurring goal of enjoying coffee. The algorithm terminates having confirmed the expectation that John would try to subsume his goal of enjoying hot coffee.

The triggered expectation adds this information about subsumption into the story representation. That is, it adds to the representation the inferences that John wanted to buy a thermos in order to keep coffee hot so that he could enjoy drinking the coffee when he so desired. This information appears in the Goal Subsumption Source structure shown in Figure 4.

### 5.2.1 Computer Example

This section contains a trace PAM understanding a Goal Subsumption Establishment story:

> (6) John got tired of going to the local singles' bars every evening. He decided to get married.

Some of the rules needed to understand (6) are different that the ones used in (5), but the basic goal subsumption recognition algorithm is the same as that described in the above example.

[PHOTO: Recording initiated Thu 10-Aug-78 8:58PM]

@RU PAM

*(UNDERSTAND CD8)

THE STORY IS

JOHN GOT TIRED OF GOING TO SINGLES' BARS EVERY NIGHT.
HE DECIDED TO GET MARRIED.

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(JOHN GOT TIRED OF GOING TO<br>SINGLES'S BARS EVERYNIGHT) | |
| CONCEPTUALIZATION IS:<br>((ACTOR HUMO <=> (*PTRANS*)<br>OBJECT HUMO TO (*PROX* PART ORGO))<br>MANNER ((*FREQUENTLY*))<br>TIME (FORM7)) | PAM breaks the first sentence into two conceptualizations. This is the first one, which denotes that John goes to singles' bars regularly. The second conceptualization denotes that he tired of this. |
| NOT A PREDICTED INPUT | |
| BEGIN SEARCH FOR EXPLANATION | PAM now tries to infer the reason someone may go to a singles' bar. |
| TESTING EXPLANATION OFFERED BY<br>UNK-DPROX-PLAN-REQ | |
| EXPLANATION IS GOAL:<br>(*DPROX* PLANNER HUMO OBJECT HUMO<br>LOCATION (*PROX* PART ORGO))<br>PLAN: (*UNSPEC* PLANNER<br>HUMO) | PAM has a rule that states that if a person goes somewhere, then he wanted to be at that place. Thus this input causes PAM to infer that John wanted to be at the singles's bar. Now an explanation for this goal is sought. |
| NO PREDICTION CONFIRMED | |
| ASSUMING EXPLANATION<br>CONTINUING SEARCH | |
| TESTING EXPLANATION OFFERED BY<br>USE-PICK-UP-PLACE-REQ | Stored under "singles' bar" in memory is that it serves as a place to find partners to have sex with. Thus PAM infers that John must go to singles' bars in order to have sex. |
| EXPLANATION IS GOAL:<br>(*SSEX* PLANNER HUMO PARTNER (NIL))<br>PLAN: ($SEX PLANNER HUMO PARTNER<br>(NIL))<br>SOURCE: (*SATISFY-DRIVE*) | |

```
EXPLANATION CONFIRMS
PREDICTION FREQUENT-EVENT-REQ

FOUND EXPLANATION SEQUENCE:

UNK-DPROX-PLAN-REQ ->
USE-PICK-UP-PLACE-REQ

*** ADDING TO STORY REPRESENTATION:
```

PAM has requests initialized before reading the story that look for various kinds of introductory information. One such kind of information is a repeated episode, such as the one inferred from this input. Repeated episodes are introductory in that they refer to goals that are not currently active in the story. That is, John does not necessarily have a Satisfy-Sex goal at this point in the story.

```
 OUTCOME OF GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION
(*PROX* PART ORGO))   IS (*SUCCEED*)
```

PAM now asserts that John goes to singles' bars to satisfy his sex urge.

```
 INFERRED GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*PROX*
PART ORGO))

 INFERRED GOAL: (*SSEX* PLANNER HUMO
PARTNER (NIL))

REPEATED GOAL EPISODE CONSTITUTES
BACKGROUND INFORMATION
```

This message indicates that PAM has treated this input as background rather than as an actual event. For example, PAM will not look for John to try to pick someone up next since the input does not imply that he has an active goal.

```
CONCEPTUALIZATION IS:
((ACTOR HUMO TOWARD
 (*FONDNESS* VAL (-5.)
 OBJECT
 ((ACTOR HUMO <=> (*PTRANS*)
   OBJECT HUMO
   TO (*PROX* PART ORGO))
  MANNER (*FREQUNETLY*))
  TIME (FORM28))))
 TIME (FORM29))
```

The second part of the conceptualization that PAM divides the first sentence into denotes that John tired of going to bars.

```
NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
RESUBSUM-REQ
```

```
EXPLANATION IS GOAL:
(*SUBSUM* PLANNER HUMO GOAL
(*SSEX* PLANNER HUMO PARTNER (NIL)))
 WITH SOURCE (*DISSATISFACTION*
INITIATOR CON233)
```

PAM infers that John may want to subsume Satisfy-sex due to his dissatisfaction with the way he is an implementation of Rule GS2 shown in the next section.

```
EXPLANATION CONFIRMS
PREDICTION INIT-REQ

FOUND EXPLANATION SEQUENCE:

RESUBSUM-REQ


*** ADDING TO STORY REPRESENTATION:

LOADING PREDICTION
RESUBSUM-AT-REQ
```

PAM loads a prediction that John will attempt to establish a subsumption state for his Satisfy-sex goal.

```
NEXT INPUT IS:
 (JOHN DECIDED TO GET MARRIED)

CONCEPTUALIZATION IS:
((ACTOR HUMO <=> (*PLAN*)
  PLAN ((ACTOR HUMO
         <=> ($MARRY PARTNER HUMO
                     PARTNER (NIL)))
         TIME (FORM37)))
 TIME (FORM38))


NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
MAKE-UP-MIND-REQ
```

PAM infers that John made a decision to do something,

```
NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
WED-PLAN-REQ

EXPLANATION IS
 PLAN: ($MARRY PLANNER HUMO PARTNER
(NIL))
```

which was to becomes married to someone.

| | |
|---|---|
| EXPLANATION CONFIRMS<br>PREDICTION RESUBSUM-AT-REQ<br><br>FOUND EXPLANATION SEQUENCE:<br><br>MAKE-UP-MIND-REQ -> WED-PLAN-REQ | The prediction that John would subsume Satisfy-sex is confirmed. |
| *** ADDING TO STORY REPRESENTATION: | |
| INPUT CONFIRMS PREDICTION<br>RE-SUBSUM-AT-REQ<br><br>FINISHED UNDERSTANDING PHASE | PAM adds to the story representation that the desire to subsume Satsify-sex gave rise to the goal of being married. |

---

[PHOTO: Recording terminated Thu 10-Aug-78 9:07PM]

### 5.2.2 Some More Rules

The previous computer example required a rule to determine that a character wanted to establish a subsumption state. This section describes a number of rules relevant to Subsumption State Establishment.

For example, the rule used in above example was the following:

Rule GS2:

If

a character has a negative attitude toward an action that may be dominated by a recurring goal,

then

that character may want to subsume the recurring goal using a plan that does not involve the distasteful action.

The application of this rule is similar to Rule GS1 above: A negative attitude toward an action, like disliking going to singles' bars, is observed, and the rule gives rise to an expectation looking for the goal of establishing a subsumption state. Then the Subsumption State Recognition algorithm is used to determine if the next input or an inference from it could be a subsumption state. Since being married subsumes Satsify-sex and Enjoy-company without involving going to a singles' bar, the understander infers that John wanted get to married to subsume these goals. The computer example in the last section shows how this story is understood by PAM.

There are rules analogous to GS1 and GS2 but which involve beliefs rather than attitudes:

Rule GS3:

If

a character has a positive belief about an action that may be dominated by a recurring goal,

then

that character may want to subsume that goal.

Rule GS4:

If

a character has a negative belief about an action that may be dominated by a recurring goal,

then

that character may want to subsume the recurring goal using a plan that does not involve the distasteful action.

For example, consider

(7) John thought that jogging would keep him healthy. He bought a pair of running shoes.

The inference must be made in (7) that John may want to subsume a P-health goal, because staying healthy is something that is good for someone, and owning jogging shoes makes it easier to execute a plan for this goal. The process of making this inference is essentially the same as that needed for the previous rules, except that here the understander needs knowledge about which states are "good" or "bad" for a person.

Another way the goal of subsuming a recurring goal may be inferred is from the statement of <u>inadequacy</u>. For example, consider

(8) John could never find three other people who wanted to play bridge. Then he read an ad for a bridge club in the newspaper.

Here a reader needs to infer that John will want to join the bridge club because it would make it easier to find other players. To understand that John will want to subsume his need for other bridge players, the following rule is needed:

Rule GS5:

If

a character has repeated <u>difficulty</u> fulfilling a recurring goal,

then

that character may have the goal of establishing a subsumption state that eliminates the difficulty.

In story (8), John had difficulty finding bridge partners. According to rule GS5, John may wish to establish a state eliminates this difficulty. Since being a member of a bridge club makes partners readily available, a reader could infer that John would want to join the club based on the inference made from Rule GS5.

This rule is different from the other rules in this section only in that it is invoked upon encountering a difficulty for a character. To apply this rule, then, we need a rigorous definition of the intuition of difficulty. Difficulty is defined in Chapter 12.

5.3 Goal Subsumption State Replacement

A variation on Subsumption State Establishment occurs when the character replaces an old subsumption state by a new one. For example, consider

(9) John got tired of riding his bicycle to work every day. He decided to buy a car.

(9) is very similar to (6) above, which was an instance of a Goal Subsumption State Establishment situation:

(6) John got tired of going to the local singles' bars every evening. He decided to get married.

As in (6), Rule GS2 is need in (9) to infer that John is purchasing a car to subsume his recurring goal of being at work.

The difference between these two stories is that in (9), an old subsumption state has been replaced. One of the reasons John may have had for owning the bicycle was to be able to ride to work each morning. Now that having a car will presumably replace this plan with driving to work each morning, John may no longer need the bike. For example, if John sold his bicycle after (9), a reader might infer that it was because he now no longer needed the bicycle. However, if John sold his furniture after (9), the reader would not make this inference. He might assume instead that John needed some money to pay for his car. Owning a car may make owning a bicycle obsolete, but it does not replace the need for furniture.

To understand that an old subsumption state may have become obsolete by the creation of a new one, the understander must keep track of how each character in a story subsumes his goals. Whenever a reader learns that some character has subsumed a recurring goal by the establishment of some state, the reader must store this state along with the recurring goal it subsumes in a list of the subsumption states that that character has achieved. In addition, when a new subsumption state replaces a old one, the old one must be marked as obsolete, i. e., tagged as no longer subsuming its recurring goal. Along with this tag should be a pointer to the new state that now subsumes the goal. In this manner, the history of the subsumption of a recurring goal gets recorded.

This history would be used in the following case. Suppose story (9) were in fact followed by

(10) John bought a new car and sold his bicycle.

This history of the subsumption of John's goal of being at work now looks like this:

```
JOHN

 Recurring-Goal:

     DPROX
      Planner     = JOHN
      Destination = LOCATION(work)

     Subsumption State: BICYCLE IS OWNED(JOHN)

     Status: Obsolete

                 New State:
                  Subsumption State: CAR IS OWNED(JOHN)
                  Status:            Active
```

That is, this goal used to be subsumed by John owning a bicycle, but now it is subsumed by John owning a car. If the reader were now asked the question

Q) Why did John sell his bicycle?

a reasonable answer would be

A) He no longer needed it to drive to work once he bought a car.

To produce this answer, the reader must have inferred that one of the reasons John sold the bicycle was that he no longer needed it. Then the reader could return this reason as an answer, stating that the object was no longer needed for the recurring goal, because of the existence of a new subsumption state.

To recognize that one subsumption state replaces another, the following algorithm is needed:

1. Compare the new subsumption state to those on the character's "subsumed goals" list, looking for one with the same goal. If none is found, then the state does not replace a previous subsumption state.

2. Take the plan whose precondition is fulfilled by the new subsumption state, and compare it against the plans used in the old subsumption states. If the new state subsumes the same goals as the old state did, but subsumes them through a different plan then the new state is replacing the old one.

The point of this recognition algorithm is to distinguish between replacement states and multiply subsumed goal states. For example, in story (10), suppose John also bought a book of tickets for the highway tolls. Both having the ticket book and having the car subsume John's recurring "be at work" goal. But they do not compete with each other because the plans each state suggests are consistent with each other.

In (10), on the other hand, the above algorithm would work as follows:

1. The new state of owning a car is compared against John's other subsumption states, and it is found to share the same subsumed goal as owning a bicycle, namely, John's goal of being at work.

2. The plans for which these states fulfill a precondition are compared. In one case, we have the plan USE-VEHICLE(JOHN, BICYCLE), and in the other, USE-VEHICLE(JOHN, CAR). Both of these plans are for the goal of being at work, and are different plans. Hence the state of owning a car replaces the subsumption state of owning a bicycle.

## 5.4 Goal Subsumption State Termination

The most interesting situation in which goal subsumption is involved occurs when a subsumption state is terminated. For example, consider the following story:

(11) John and Mary were happily married for a number of years. Then one day, John was killed in a car accident.

The reader of (11) must realize that John being killed terminates the marriage subsumption state for Mary. Being married normally subsumes a number of goals for the participants, that must now be dealt with individually. For example, a reader might infer that Mary became lonely, and had trouble paying the rent and looking after the kids.

The reader might also infer that she wanted to subsume these goals again. For example, suppose (11) were followed by

(12) Mary had to go to work.

If the reader were asked the question

Q) Why did Mary go to work?

an appropriate answer would be

A) Mary had to support herself now that John was gone.

Providing this answer requires the reader to know that having a job can subsume one of the same economic goals that marriage can. Since the marriage state no longer exists to subsume these goals, the reader can

infer that Mary might be looking for another way of subsuming them.

Finding this interpretation of (12) required the following inference rule:

Rule GS6:

If
a character has a subsumption state that terminates,

then
that character may want to establish another subsumption state for the same goals.

This rule is similar to the rules for Goal Subsumption State Establishment in that it can be used to predict that the character will seek to subsume these goals. When an attempt is made by the character to subsume these goals, the understander can use the Goal Subsumption Recognition algorithm defined in the section on Goal Subsumption State Establishment in order to interpret the story.

Another rule related to the termination of a goal subsumption state is the following:

Rule GS7:

If
a character has a subsumption state that terminates,

then
that character may fail to fulfill a precondition for the recurring goal when it arises.

Rule GS7 is needed for a story like

(13) After John lost his job, he became hungry.

To answer the question

Q) Why did John become hungry?

with

A) Because he couldn't afford enough food after he lost his job.

the understander must infer that losing one's job terminates that subsumption state. Since having a job subsumes economic goals, then by rule GS7, John may not be able to achieve some goals that require possessing an object. Satisfying hunger is such a goal, because the plan for it requires the possession of food. Thus the understander infers that John was hungry because John couldn't afford food because his stream of income was cut off.

Rules GS6 and GS7 are rules about what might happen after a goal subsumption state is terminated. In addition, the reader requires a set of rules to determine that a subsumption state is in fact terminated. These rules include the following:

Rule GS8:

If

a state is terminated,

then

that state is terminated as a subsumption state.

By this rule I mean that when a reader learns of the termination of a state, he must then check to see if the characters involved in that state have a goal subsumed by it. If they do, then that subsumption state should be marked as terminated. For example, if John sells his car, he no longer owns it, so it can no longer subsume his transportation needs. If John divorces Mary, a subsumption state is terminated for both of them, since both of them are involved in the marriage state.

Rule GS9:

If

1. a character dies,

and

2. that character was in some social relationships with other people,

then

those social relationships are terminated.

This was the case in story (11). John's death terminated his marriage to Mary according to Rule GS9, and according to Rule GS8, this ended a subsumption state for her.

Rule GS10:

If

a functional object is lost or stolen,

then

the subsumption states created by its ownership are terminated.

GS10 is necessary because a person who loses an object or has it stolen from him still owns it. Thus the relationship of ownership may still exist, but it is inadequate as a subsumption state.

### 5.4.1 Computer Example

The following is a trace of PAM understanding a version of the story

(14) John and Mary were happily married for a number of years. Then one day, John was killed in a car accident. Mary had to get a job.

[PHOTO: Recording initiated Thu 10-Aug-78 9:01PM]

*(UNDERSTAND CD10)

THE STORY IS

JOHN AND MARY WERE HAPPILY MARRIED FOR A NUMBER OF YEARS.
THEN ONE DAY, JOHN WAS KILLED IN A CAR ACCIDENT.
MARY HAD TO GET A JOB.

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(JOHN AND MARY WERE MARRIED) | |
| CONCEPTUALIZATION IS:<br>((IS (*MARRIED* PARTNER HUM0<br>PARTNER HUM1)) TIME (FORM11)) | |
| NOT A PREDICTED INPUT | |
| BEGIN SEARCH FOR<br>EXPLANATION | |
| TESTING EXPLANATION OFFERED BY<br>SOC-REL-REQ<br><br>INPUT IS BACKGROUND INFORMATION<br>INFORMATION | PAM finds that the input is the statement of the existence of a social relationship, which is treated as background information. That is, the information that the relationship exists is just stored as a fact about these characters. |
| EXPLANATION CONFIRMS PREDICTION<br>BACKGROUND-REQ | |
| FOUND EXPLANATION SEQUENCE: | |
| SOC-REL-REQ | |

```
*** ADDING TO STORY REPRESENTATION:

ASSERTING *MARRIED* RELATIONSHIP

SOCIAL RELATIONSHIP ENCOUNTERED


NEXT INPUT IS:
 (JOHN WAS KILLED IN A CAR ACCIDENT)

CONCEPTUALIZATION IS:
((CON
   ((ACTOR PHYS0 <=> (*PROPEL*)
     OBJECT HUM0)
    MANNER ((*UNINTENTIONAL*))
    TIME (FORM12))
   LEADTO
   ((ACTOR HUM0 TOWARD
     (*HEALTH* VAL (-10.)))
  TIME (FORM13))))
```

The input stated that the car hit John accidently, so PAM does not try to find an an explanation for it. That is, the program knows not to look for intentional explanation for something that is marked as unintentional. Instead, PAM just infers the consequences of the event.

```
NOT A PREDICTED INPUT


BEGIN SEARCH FOR EXPLANATION


TESTING EXPLANATION OFFERED BY
DEATH-REQ
```

PAM finds a rule stating that the character's social relationships should be marked as terminated.

```
FOUND EXPLANATION SEQUENCE:

STRIP-CAUSE-REQ -> DEATH-REQ


*** ADDING TO STORY REPRESENTATION:

RELATIONSHIP *MARRIED* IS TERMINATED
```

PAM marks John and Mary's marriage relationship as being terminated.

```
NEXT INPUT IS:
 (MARY HAD TO GET A JOB)
```

```
CONCEPTUALIZATION IS:
((CON
  ((ACTOR HUM1 IS (*EMPLOYED*))
   TIME (FORM17))
  IS (*GOAL* PART HUM1))
 TIME (FORM18))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
AJOB-REQ

EXPLANATION IS GOAL:
(*AJOB* PLANNER HUM1 JOB (NIL))

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
SUBSUME-GOAL-REQ
```

PAM finds a request that interprets Mary's goal as being generated by the need to subsume her goal of having money. This inference is based on the facts that the termination of a subsumption state can give rise to the goal of establishing a new subsumption state, and that marriage can subsume goals involving money.

```
EXPLANATION IS GOAL:
(*AJOB* PLANNER HUM1 JOB (NIL))
 WITH SOURCE
(*GOAL-SUBSUMPTION*
 RGOAL
  (*DCONT* PLANNER HUM1 OBJECT
   TOK1 OWNER (NIL) RECIPIENT HUM1)
 INITIATOR
  ((ACTOR HUM0 TOWARD (*HEALTH*
    VAL (-10.))) TIME (TIMK3))

EXPLANATION CONFIRMS PREDICTION
INIT-REQ

FOUND EXPLANATION SEQUENCE:

AJOB-REQ -> SUBSUME-GOAL-REQ


*** ADDING TO STORY REPRESENTATION:

INFERRED GOAL: (*AJOB* PLANNER HUM1
JOB (NIL))
```

The inference that Mary was trying to subsume her need for money is added to the story representation.

```
INFERRED GOAL SOURCE:                     |
 SUBSUME GOAL: (*DCONT* PLANNER HUM1      |
OBJECT TOK1 OWNER (NIL) RECIPIENT         |
HUM1)                                     |
 INFERRED INITIATOR:                      |
((ACTOR HUMO TOWARD (*HEALTH* VAL         |
(-10.))) TIME (TIMK3))                    |
                                          |
FINISHED UNDERSTANDING PHASE              |
```

QUESTION: Q1
Why did Mary need employment?
John died and so she needed a source of money.

[PHOTO: Recording terminated Thu 10-Aug-78 9:03PM]

To answer the question above, PAM looks at the source of Mary's goal of having a job. PAM has inferred that the source was the need to subsume her financial needs, and that this was initiated by John's death. Thus PAM use this information to explain Mary's need.

### 5.5 Other Situations Involving Goal Subsumption

In addition to these three story types, goal subsumption is also important in the creation of goal conflicts. For example, consider

(15) John loved his wife Mary very much, but their sex life was terrible.

Here the marriage relationship subsumes a number of goals, but not all of them adequately. It subsumes John's desire to be with Mary, an inference from his loving her, but it fails to fulfill his sexual needs. Thus John may want to terminate the relationship in order to better fulfill his sexual needs, but he may also want to maintain the relationship because the other goal subsumed by the marriage are deemed more important. Thus the marriage goal subsumption state has created a goal conflict for John. This situation is studied in more detail in Chapters 6 and 7 on Goal Conflict.

### 5.6 Summary

Knowledge about goal subsumption is needed to understand story situations in which a goal subsumption state is established, when an existing goal subsumption state is replaced by a new subsumption state, or when a goal subsumption state is terminated.

The following rules were found useful for processing these situations:

Rule GS1:

If

a character has a positive attitude toward performing an action,

then

that character may want to establish a state that subsumes the goal dominating that action.

Rule GS2:

If

a character has a negative attitude toward an action that may be dominated by a recurring goal,

then

that character may want to subsume the recurring goal using a plan that does not involve the distasteful action.

Rule GS3:

If

a character has a positive belief about an action that may be dominated by a recurring goal,

then

that character may want to subsume that goal.

Rule GS4:

If

a character has a negative belief about an action that may be dominated by a recurring goal,

then

that character may want to subsume the recurring goal using a plan that does not involve the distasteful action.

Rule GS5:

If

a character has repeated difficulty fulfilling a recurring goal,

then

that character may have the goal of establishing a subsumption state that eliminates the difficulty.

Rule GS6:

If

a character has a subsumption state that terminates,

then

that character may want to establish another subsumption state for the same goals.

Rule GS7:

If

a character has a subsumption state that terminates,

then

that character may fail to fulfill a precondition for the recurring goal when it arises.

Rule GS8:

If

a state is terminated,

then

that state is terminated as a subsumption state.

Rule GS9:

If

1. a character dies,

and

2. that character was in some social relationships with other people,

then

those social relationships are terminated.

Rule GS10:

If

a functional object is lost or stolen,

then

the subsumption states created by its ownership are terminated.

# CHAPTER 6

# GOAL CONFLICT

Pursuing one goal may make it difficult for a character to fulfill some other goal. If a character has a number of goals that interfere with one another, then that character is said to have a goal conflict. An understander needs to recognize goal conflicts when they occur in stories in order to understand the behavior of a character whose goals are in conflict.

Recognizing goal conflicts requires that the reader know how goals can conflict. To fulfill the requirements of this task, the following classification of goal conflicts is proposed: There are goal conflicts caused by resource limitations, by mutually exclusive states, and by the generation of a preservation goal. For example, a limitation of resources can cause a goal conflict if the plans for a character's goals require more time than is available to the character, or if the plans require more of a consumable resource, such as money, than is possessed by the character. Mutually exclusive states can give rise to goal conflicts if the goal states are exclusive, as in wanting to be married to two people at once. The generation of a preservation goal can cause a goal conflict if a plan for a character's goal entails the creation of a preservation goal that must then be dealt with. For example, wanting to kill someone you hate may cause a goal conflict since murder usually has the undesirable side-effect of invoking the wrath of the authorities.

This chapter presents a classification system for goal conflicts and shows how it may be used to detect them. The next chapter discusses story situations in which goal conflicts can occur. In that chapter, the goal conflict classification is used to understand the kinds of stories to which goal conflicts can give rise.

## 6.1 Introduction

People often have more than one goal at a time. To understand a story in which a character has several goals, the reader must follow the course of each goal. However, a character's goals cannot always be followed independently of each other. The reader must understand how a character's goals interrelate in order to understand that character's behavior.

For example, consider following the stories:

(1) John wanted to watch the Monday night football game. He also wanted to have steak for dinner. On his way home from work, John stopped at the butcher's. Later that evening, he watched the football game.

(2) John wanted to watch the Monday night football game. He also had a paper due the next day. That night, John watched the football game. John failed Civics.

(3) John wanted to marry Mary. He also wanted a new car. John proposed to Mary. Then he went to see a car salesman.

(4) John wanted to marry Mary. He also wanted to marry Sue. John took Mary out and proposed to her. She agreed. Later, John called Sue and told her he wouldn't be seeing her anymore.

John has two goals in each of the above stories. In stories (1) and (3), a reader could explain John's behavior by following each goal independently. John watched the football game to fulfill one goal in story (1), and stopped at the butcher's in pursuit of a second. Proposing to Mary was part of a plan for marrying Mary in story (3), and going to a car salesman was part of a plan for getting a car. In fact, both of these stories can be made into two understandable stories by separating the events associated with each goal. Story (1) could be divided into these stories:

(5) John wanted to watch the Monday night football game. That night, John watched the football game.

(6) John wanted to have steak for dinner. On his way home from work, John stopped at the butcher's.

Story (3) would become:

(7) John wanted to marry Mary. He proposed to her.

(8) John wanted a new car. John went to see a car salesman.

In contrast, stories (2) and (4) cannot be understood by following each goal separately. There is no intrinsic reason that watching a football game should cause one to fail a course, as it did in story (2). Nor can one explain why John no longer wanted to see somebody he was considering marrying in story (4). If these stories were divided by separating the events having to do with each goal, the results are not very sensible. Story (2) would become the following two stories:

(9) John wanted to watch the Monday night football game. That night, John watched the football game.

(10) John had a paper due the next day. John failed Civics.

and story (4) becomes:

(11) John wanted to marry Mary. John took Mary out and proposed to her.

(12) John wanted to marry Sue. John called Sue and told her he wouldn't be seeing her anymore.

Stories (9) and (11) are coherent, but stories (10) and (12) are not. The reader might infer that John didn't write his paper in story (10), but can find no explanation of why he didn't. Similarly, it is hard to explain why John stopped seeing Sue in story (12).

However, in the original versions of these stories, the explanations for these events are readily available: In (2) John spent too much time watching the football game, so he didn't have enough time left to write his paper. In (4) John decided to marry Mary, so he had to give up seeing other women. Even though the goals in each story are stated as seemingly separate goals, these explanations are not found when the goals are considered separately in stories (10) and (12).

The key problem in each of these stories is inferring the connection between the story's goals. For example, the connection between the goals in story (2) is based on a shortage of available time. The goals of story (4) are connected because being married to someone is usually assumed to prohibit additional romantic involvements. Since these connections are not explicit in the stories, a reader has to infer them in order to explain the characters' behavior. Thus a reader needs to know what kinds of connections there may be among a character's goals, how these connections can be inferred, and what the consequences of each connection are.

### 6.1.1 Goal Conflict

The interrelationship of the goals in story (2) and those in story (4) is called goal conflict. A goal conflict is a situation with the following two components:

1. A character has several goals at the same time.

2. The fulfillment of one of these goals interferes with the fulfillment of the others.

For example, in story (2) John has the goal of enjoying the football game, and the goal of writing a paper. These goals are in conflict because there may not be enough time to fulfill both of them. In story (4), John's two goals are "being married to Mary" and "being married to Sue." These goals conflict because a person cannot be married to two different people at the same time.

In story understanding, a reader must detect goal conflicts if he is to understand a character's behavior with respect to those conflicting goals. For example, suppose that after reading story (2), a reader were asked the question

Q1) What would have happened if John hadn't watched the football game?

To respond with the answer

A1) John might have passed Civics.

the reader has to infer that John failed Civics because he didn't write his paper. To infer that John didn't write his paper, the reader has to realized that writing the paper conflicted with watching a football game. If a character has a goal conflict, and fulfills one of the conflicting goals, an inference is that he will fail to fulfill the others. Since John fulfilled his goal of watching the game, he must have failed to fulfill his goal of writing the paper.

Answering question Q1 requires the reader to have spotted the conflict between John's goals. Then knowledge about goal conflicts could be used to explain why John failed his course. Thus a story understander must be able to detect goal conflicts and infer their consequences. In this chapter, I discuss the various kinds of conflicts that can occur between goals, and give algorithms for detecting them. The next chapter is concerned with understanding a story once the interrelationships of that story's goals have been inferred.

## 6.2 Detecting Goal Conflicts

To understand a story in which a goal conflict explains a character's behavior, a reader must first recognize that the goal conflict exists. Recognizing a goal conflict requires the reader to compare each goal it learns about against the known goals of the same character. This comparison presumes that the reader knows about the various ways goals can conflict with one another, and can use this knowledge to determine whether the goals it is comparing are in conflict.

Thus a story understander must have knowledge about how goals can conflict. This section contains a classification of goal conflicts that is based upon similarities between the kinds of knowledge required to spot each goal conflict situation. An understander supplied with this knowledge could use it to determine if a goal conflict situation exists between the goals it is comparing.

### 6.2.1 Kinds of Goal Conflicts

I found it necessary to distinguish three classes of goal conflicts. They are as follows:
Types of Goals Conflicts

1. Resource Limitations

Two goals can conflict if the plans chosen for those goals share a common resource, and there is an insufficient quantity of that resource available for both plans. Consider the following story:

(13) John's friends asked him to go bowling. John had promised Mary he would take her out to dinner, but his friends persuaded him to come with them. The next morning, Mary told John she was going to divorce him.

The scarce resource in story (13) is time. To determine that there is a goal conflict due to a scarcity of a resource in this story, a reader needs to know what resources are required by each of the character's plans, and how much of that resource is available to him. Since all plans require some amount of time to be executed, a goal conflict based on time limitations is always a possibility for a character with more than one goal. To spot the time scarcity in story (13), a reader needs to recognize that John's plans have time restrictions on them. These restrictions constrain John to execute both plans in the same time period. Since it is unlikely that both plans can be executed concurrently, a reader can compute that John has a goal conflict based on a limitation of time resources.

2. Mutually Exclusive States

Two goals can also be in conflict if the states they are intended to achieve are incompatible. An example is the following:

(13) John wanted to move to California, but he also wanted to live near his relatives in Kansas.

Fulfilling each goal requires John to be living in two different places. Since being in two different places at once is not possible, John has a goal conflict.

To recognize this conflict, a reader must be able to compute the conditions that will be brought about in the pursuit of a goal, and must have a way of determining if they are compatible. In example (13), these conditions are living in California and living in Kansas. These conditions can be computed because they are the conditions that John's goals are designed to bring about. To determine that they are exclusive, the reader must use knowledge about the physical world. In particular, the reader must know that a person cannot be in two locations at once unless the locations overlap. Kansas and California do not, so John cannot fulfill both his goals.

3. Causing a Preservation Goal

A character may have a goal conflict if executing the plan for a goal will cause a preservation goal to come into being. For example, consider the following story:

> (14) John wanted to go to the football game, but it was raining outside.

To fulfill his goal of going to the football game, John would have to be outside in the rain. Being in the rain would cause John to become damp, and being damp is an undesirable state. Thus going to the football game would cause John to have the goal of preventing himself from getting damp. Story (14) therefore contains a form of goal conflict because John's goal of being at the football game would make it difficult for him to filfill the preservation goal of maintaining his comfort.

This form of goal conflict differs from the others in that the character does not have two goals to begin with. Instead, the character has one goal whose plan would generate a goal that the character may have difficulty fulfilling. For example, in story (14), John has a goal conflict because his plan to satisfy his goal of going to the football game would generate the preservation goal of maintaining his comfort.

Thus a reader must be able to detect this type of goal conflict upon seeing only one goal. The reader needs to examine the plan for that goal, and determine if its execution violates any state that the planner is likely to want to preserve. For example, in story (14), the rain causes a preservation goal only because John has another goal that requires him to go out in the rain. The reader must infer a preservation goal in this case, but not in the following story

> (15) John loved to listen to the sound of the rain. It was raining outside.

Here the rain is perfectly compatible with John's goal, so the reader should not infer a goal conflict.

Whenever a reader learns of a new goal for a character, the reader must see if the goal causes any of these three kinds of conflict. The general form of the algorithm for goal conflict detection is to ask the questions in Figure 5. Each of these goal conflict situations is now described in detail.

### 6.2.2 Resource Limitations

A goal conflict situation caused by a limitation of resources has the following features:

1. A set of goals, each with a plan.

2. A common resource required by each of these plans.

3. An insufficient quantity of this resource available.

A resource is something needed to perform a plan. For example, the following stories contain goal conflicts caused by a shortage of resources:

(16) John had just enough money to buy either a television or a stereo, but he wanted to have both.

(17) John wanted to watch the football game, but he had a paper due the next day.

(18) John wanted to cook linguine and hamburgers on his camp stove, but the stove had only one burner.

In story (17), watching the football game and writing a paper each require a certain amount of time, and the amount of time available to John is limited because the paper is due in a day. In story (16) John has two goals that are ordinarily achieved by purchasing the desired items. Purchasing an object requires that one has money, but the amount of money needed for both these goals exceeds John's means. John's plans for the goals in story (18) each require the use of a burner, but John can only use his burner for one goal at a time.

Recognizing the goal conflict in each of these stories requires the understander to perform several processes. First, the understander has to determine that a shortage of resources is likely. Most of the time, the actual resources needed to execute a plan are not specified in the story, nor is the amount of the resource available to the planner. The reader usually can infer resource shortages from "hints" in the text.

**Figure 5**
**Overview of Goal Conflict Detection**

```
 ______________________________
|                              |
| Does the new goal require    |           _________________
| a resource also required     |  yes     |                 |
| by another goal and of       | =====>   | The new goal con-|
| which there is an insuff-    |          | flicts with an old|
| icient quantity?             |          | one.            |
|______________________________|          |_________________|

              ||
              || no
              \/
 ______________________________
|                              |
| Does the fulfillment of      |           _________________
| the new goal entail the      |  yes     |                 |
| creation of a state that     | =====>   | The new goal con-|
| is mutually exclusive        |          | flicts with an old|
| with a state needed for      |          | one.            |
| another goal?                |          |_________________|
|______________________________|

              ||
              || no
              \/
 ______________________________
|                              |           _________________
| Does the fulfillment of      |          |                 |
| the new goal entail the      |          | The new goal con-|
| generation of a preserv-     | =====>   | flicts with the |
| ation goal?                  |          | preservation goal.|
|______________________________|          |_________________|

              ||
              || no
              \/
       ____________________
      |                    |
      | The new goal does  |
      | not cause a goal   |
      | conflict.          |
      |____________________|
```

For example, story (16) suggests that a conflict due to a shortage of money may exist by explicitly stating that John's supply of money was inadequate for both goals. Story (17) indicates that time may be important by explicitly stating that one of John's goals has a deadline built into it. Story (18) hints at the goal conflict by explicitly

stating the inadequacy of the stove. Hints alone are not sufficient to spot a goal conflict, but they are useful for suggesting one's presence.

Once the understander has determined that there may be a shortage of a particular resource, it needs to confirm that this is the case. To perform this computation, the reader must infer the plans that will most likely be used to achieve each goal, and see if the resource in question is common to all the plans. The understander must also compute the total amount of the resource required by the plans, and the amount of that resource that the character has available. If more resources are required than are available, then a conflict exists.

For example, story (16) explicitly suggests that there is a shortage of money. Now the reader must try to compute the plans normally used for each goal, and determine if money is required. The plan usually used to acquire a television or a stereo is to buy one. That is, stored in memory under the goal of acquiring possession of an object is the default plan of purchasing it. Money is one of the resources required for buying things. Attached to the memory representation of the plan for buying an object is the following information:

```
BARGAIN(PLANNER,PERSUADEE,REQUEST,OFFER)
                                                   |-> PLANNER
 REQUEST = PERSUADEE <=> ATRANS <--O-- OBJECT -|
                                                   |-< PERSUADEE

                                              |-> PERSUADEE
 OFFER = PLANNER <=> ATRANS <--O-- MONEY -|
                                              |-< PLANNER

 CONSTRAINTS:

         [ MONEY IS POSS(PLANNER)

           OBJECT IS POSS(PERSUADEE) ]

 CONSEQUENCES:

         [ PLANNER IS POSS(OBJECT)

           PERSUADEE IS POSS(MONEY) ]
```

This states that the plan for buying an object is a special case of bargaining in which the planner gives the persuadee money in exchange for the persuadee giving the planner the requested object. The contraints in the diagram show that money is required to use the plan. The consequences show that money is consumed by the plan by changing possession. Since the reader was told that John didn't have enough money for both purchases, the reader can infer that John has a goal conflict based on a limitation of the resource money.

The process of detecting goal conflict due to a limited resource is diagrammed in Figure 6. Most of the difficulty in performing the algorithm in Figure 6 involves determining the amount of resources required by the plans, and the amount of that resource available to the planner. This task is problematic because the reader is not always told explicitly that a resource is scarce, and computing the amount of resources required by a plan and the amount available depends entirely on the type of resource required.

For example, in story (17), the reader is not explicitly told that there was a shortage of time. To determine if there is, the reader must made a computation that is unique to time. For example, in order to know the total amount of time required by two plans, the reader must determine if the plans could be carried out concurrently. If John were capable of writing his paper and watching the game at the same time, no goal conflict would exist in the story.

To determine if a shortage of resources existed in a multi-goal situation, I found it necessary to divide resources in several classes, depending on how the resources can be allocated. These classes are:

1. Time

   The amount of time required to carry out a plan could cause a goal conflict if the amount of time available to the planner was limited.

2. Consumable Functional Objects

   A functional object is an object that is normally used in a particular plan. A consumable functional object is one whose use to the planner is reduced when it is used. For example, money is a functional object used for trading. It is also a consumable object because it is no longer available to the planner after he buys something with it. The functional nature of objects is discussed in detail in Chapter 11.

   If the plans for a set of goals all require the use of a consumable object, then a goal conflict can result if the planner does not have a sufficient quantity of that consumable.

3. Non-consumable Functional Objects

   A non-consumable object was one that was not used up when it took part in a plan. For example, in story (18), the stove was a non-consumable functional object that was used to cook things. Non-consumable functional objects usually are involved in goal conflicts only when there is some time pressure. In story (18) a goal conflict exists because John's stove is not sufficient for his needs. However, if the requirement that John cook the two dishes at the same time were removed from story (18), no conflict would be present.

Figure 6
Detecting a Conflict Caused by a Limited Resource

```
 ___________________________________
|                                   |        ______________
| Does the story suggest a          |   no  |              |
| scarce resource?                  | ====> | No conflict  |
|___________________________________|       |______________|

                ||
                || yes
                \/
 _______________________________________________
|                                               |
| Infer the plan usually used for each goal.    |
|_______________________________________________|

                ||
                ||
                \/
 ___________________________________
|                                   |        ______________
| Is the resource common to         |   no  |              |
| each plan?                        | ====> | No conflict  |
|___________________________________|       |______________|

                ||
                || yes
                \/
 ___________________________________
|                                   |        ______________
| Is the amount available less      |   no  |              |
| than the amount needed?           | ====> | No conflict  |
|___________________________________|       |______________|

                ||
                || yes
                \/
     _____________________________
    |                             |
    | There exists a conflict due |
    | to resource limitations.    |
    |_____________________________|
```

4. **Abilities**

**An ability was the extent to which a character could perform an action. For example, a person has the ability to run only so fast and for so far, or to function at a particular intellectual level. As was the case with non-consumable objects, abilities are useful in detecting goal conflicts only when there is a time restriction present. For example, John's**

ability to pay attention to one thing at a time played a role in the goal conflict in story (17). But had there been no time limitation in that story, this same limitation on John's capabilities would not have caused a problem.

As this description of resources indicates, there are two kinds of computations required to determine if a shortage exists. One computation involves time, non-consumable objects, and abilities, and the other computation involves only consumable objects. The algorithms for each computation are now described.

6.2.2.1 Time

All plans require some amount of time to be executed. Thus a goal conflict based on a limitation of time resources is possible whenever a character has more than one goal at a time. Goal conflicts based on a shortage of time resources are usually hinted at by the occurrence of one of the following in the text of the story:

1. A statement of a deadline.

2. A statement of a limitation of a functional object.

3. A statement of a limitation of a person's abilties.

4. An explicit statement of a time shortage.

When one of these statements occurs in a story, the reader can infer that a time shortage exists. For example, story (17) contains an instance of an explicit deadline:

(17) John wanted to watch the football game, but he had a paper due the next day.

Since the story states that John's plan for writing his paper must be completed by a certain time, a reader of the story should infer that a time shortage is present.

A statement of a deadline for initiating a plan also indicates a conflict. For example, consider

(19) John wanted to chat with Mary, but his train left in five minutes.

Story (19) states that John's plan must be initiated in a few minutes if it is to succeed. Since this is a deadline on plan initiation, the reader should infer a goal conflict based on a shortage of time resources.

Inferring the goal conflict in each of these stories can be done with the following rule:

Rule T1:

If

1. A character has two or more goals,

and

2. the story specifies that the plan for one of the goals must be initiated, completed, or occurring at or before a particular point in time,

then

infer that a goal conflict based on a shortage of time exists among that character's goals.

Story (18) above contains an example of a goal conflict hinted at by a statement of the limitation on a functional object:

(18) John wanted to simmer two dishes on his camp stove, but the stove had only one burner.

Here the story explicitly mentions that a functional object has a limited ability. Thus the reader should infer that a time shortage based on the lack of capacity of a functional object. This is done in accordance with the following rule:

Rule T2:

If

1. A character has two or more goals,

and

2. the story specifies that there is a limitation on a functional object used by the plans for both goals,

then

a goal conflict based on a shortage of time and a limitation of a functional object exists among that character's goals.

The limitations of a person as well as an object can indicate a goal conflict. For example, consider the story

(20) John had to write a paper for a course, and read a book for an exam, but John was a very slow reader.

Since the story states that John's reading ability is below normal, the reader can infer that he will have difficulty fulfilling both goals. This conforms to the following rule:

Rule T3:

If

1. A character has two or more goals,

and

2. the story specifies that there is a limitation on the ability of the planner to perform a plan for one of those goals,

then

a goal conflict based on a shortage of time and a limitation of abilities exists among that character's goals.

The following story is an example of a goal conflict in which the time shortage is explicitly stated:

(21) John wanted to visit both Fred and Bill, but he didn't have enough time to see them both.

This story states that John had a shortage of time, so the presence of a goal conflict is certain. This is summarized in Rule T4:

Rule T4:

If

1. A character has two or more goals,

and

2. the story specifies that the character does not have enough time to execute the plans for all the goals,

then

infer that a goal conflict based on a shortage of time exists among that character's goals.

One additional way a time-based goal conflict can be detected is when a story mentions that a character has several goals, pursues one of them, and the others fail. For example, consider the following variation on story (17):

(22) John wanted to watch the football game. He also had to write a paper for a course. John watched the football game. John failed Civics.

Unlike story (17), story (22) does not explicitly mention a deadline by which the paper must be completed. However, most readers can infer that there probably was a goal conflict based on time resources in this story. This inference is made with the following rule:

Rule T5:

If

1. A character has two or more goals,

and

2. the story specifies that the character pursued one of these goals,

and

3. the character failed to fulfill the other goals,

and

4. the plans for these goals do not require any common consumable resources,

then

infer that a goal conflict based on a shortage of time exists among that character's goals.

In story (22), the plans for John's goals do not share any common resources. That is, watching a football game does not consume any objects, and writing a paper consumes only paper and ink. Therefore Rule T4 states that a goal conflict based on time must have existed. This rule is a way of stating that a shortage of time is a default reason for failing to fulfill a goal when the plans for those goals are otherwise unrelated.

Thus when a reader sees one of these hints for the presence of a goal conflict, the reader should infer that a goal conflict is present. Of course, this need not be the case. For example, consider the following story:

(23) John wanted to chat with Mary, but his train left in two days.

Story (23) seems to suggest that a time shortage is present, but then goes on to dispel this misconception. Since John's train does not leave for two days, there should be plenty of time in which to chat with Mary. However, Rule T1 above will infer a conflict to be present in this story.

The problem here is that story (23) is an unusual story. It is hard to imagine what its author was trying to convey when the story was written. For pathological stories, the inference rules given above may find a goal conflict where none is present.

Most of these pathological situations seem to be detectable with simple consistency checks. For example, after using one of the above rules to hypothesize a time-based goal conflict, the reader can check to see there is explicit information contradicting this inference. That is, the reader tries to determine if the time available is more than the time required by the plans. For this task, the reader needs to know the approximate amount of time required for a plan. In story (23), chatting usually takes much less time than is available before John must catch his train. Since this contradicts the finding of Rule T1, the story will be deemed unreasonable.

There are two reasons for separating the goal conflict identification process into one component that infers the presence of the conflict, and another component that checks for consistency. First, I am interested in understanding natural stories, and for this task the first component seems to be sufficient (That is, I have been able get PAM to process goal conflict stories by inferring the conflict without first trying the consistency checks). Second, stories usually do not specify enough information to determine if a goal conflict is truly present, but they do contain the information to detect inconsistencies. Thus separating the process into two parts enables the reader to infer conflicts based on evidence within a story, but still detect oddness when it occurs.

By inferring that a goal conflict based on a limitation of time resources exists, the story reader has made two assumptions about the plans for the character's goal. These assumptions are as follows:

1. The amount of time available to the planner to perform the plans for his goals is less than the minimum amount of time needed to execute these plans sucessfully.

2. The character is not capable of executing all the plans at the same time.

For example, recall the time-related goal conflict in story (17):

(17) John wanted to watch the football game, but he had a paper due the next day.

By inferring that a time-based goal conflict is present in this story, the reader determines that there is not enough time left for John to fulfill both goals, and that the plans for the goals cannot be carried out simultaneously. That these conditions are true of goal conflict situations is demonstrated by the following examples:

(24) John wanted to let the cat out, but he had a paper due the next day.

(25) John wanted to watch the football game, but he had a paper due next month.

(26) John wanted to sit at his desk, but he had a paper due the next day.

In each of these stories, the goal conflict present in story (17) appears to have vanished. There is no shortage of time present in story (24) because the amount of time required to fulfill the first goal has been greatly reduced. It should only take a minute to let out the cat, whereas watching the football game consumes several hours. The goal conflict disappears because the amount of time required for the goals has been substantially reduced.

In story (25), the time span within which the plans must be executed has been lengthened considerably. Even though the plans still require as much time as in story (17), there is no goal conflict in story (25) because there is enough time for the plans to be carried out sequentially.

Story (26) varies the character's ability to execute the plans. There is no goal conflict in this story because it was within the John's abilities to perform both these plans at the same time. In contrast, the plans for the goals in story (17) each required too much of one of John's resources to be carried out concurrently. Namely, both required John's attention, which could not be divided effectively between the two.

The importance of these assumptions about the planner's available time and capabilities in a goal conflict manifests itself in the kinds of situations that may occur after the goal conflict has been detected. For example, suppose that story (17) were continued as follows:

(27) John wanted to watch the football game, but he had a paper due the next day. John asked his teacher for an extension.

To understand how asking for an extension is related to John's goals, a reader must realized that John does not have enough time for both goals, and that extending the deadline on the plan for one of the goals is a way of getting more time. Thus while the rules given above work nicely for detecting the presence of a goal conflict, the understander needs to know more about the underlying reasons for the conflict in order to understand how a character may try to resolve the conflict. These factors are discussed in detail in the next chapter, which is concerned with understanding the situations in which goal conflicts arise.

6.2.2.2 Consumable Objects

A consumable object is a functional object whose use to the planner is reduced after it performs its function. For example, food is a consumable object because it can no longer be used after it is has been eaten. Money is a consumable object because it can no longer be used by a planner after he buys something with it. Consumable objects are discussed further in Chapter 11.

If the plans for a set of goals all require the use of a consumable object, then a goal conflict can result if the planner does not have a sufficient quantity of that consumable. For example, recall story (16) above:

(16) John had just enough money to buy either a television or a stereo, but he wanted to have both.

Here John has the goal of acquiring a television, and another goal of acquiring a stereo. The usual plans for both of these goals is to purchase the desired item. Buying an object requires the planner to have a certain quantity of money. Money is a consumable object, and so John would need as much money as is required by the plans for both his goals. Since the total amount of money required is greater than the amount of money John possesses, John has a goal conflict. He has to decide which object he will acquire, and which object he will do without.

The goal conflict in story (16) is recognized by the following rule:

Rule C1:

If

1. The plans for a character's goals each require the same consumable object,

and

2. The planner has enough of the consumable for any individual plan,

but

3. The total amount of that consumable owned by the planner is less than the amount needed to execute all the plans,

then

A goal conflict based on a shortage of that consumable object exists among that character's goals.

Rule C1 states that to determine if a goal conflict based on a shortage of a consumable resource exists in a story, the reader must know how much of that resource the character has available to him, and how much he needs for each of his plans. Also, the planner must have enough of the resource to fulfill each goal independently. For example, the following story should not be considered a goal conflict even though John does not have sufficient resources for both goals:

(28) John wanted to buy a television and a stereo, but he didn't have any money.

The goals in story (28) do not conflict because John could not fulfill either goal had it arisen by itself.

Some other instances of the application of Rule C1 are the following:

(29) John wanted to put up two posters, but he only had one thumbtack.

Each part of Rule C1 holds as follows:

1. The plans for each of John's goals both require the use of a thumbtack. A thumbtack is an instance of an object that is consumed by continuous enablement. That is, a thumbtack cannot be re-used without disabling its previous use. Consumption by continuous enablement is discussed in Chapter 11.

2. John had enough of the resource in question to fulfill each goal independently. Putting up a poster can be done with a single thumbtack. Since John had one thumbtack, John could put up either poster.

3. John did not have enough resources to put up both posters. Each plan requires at least one tack, so together two tacks are needed. John only has one, so a goal conflict is present.

Rule C1 is also applicable to the following story:

(30) John had to blow up two bridges, but he only had one explosive charge left.

1. The plans for John's goals both require the use of an explosive. An explosive is a physically consumable substance because it fails to exist after it is used.

2. John had enough explosives for each bridge. Presumably, one explosive charge could destroy a single bridge.

3. John did not have enough resources to blow up both bridges. Each plan requires at least one charge, so together two charges are needed. John only has one, so a goal conflict is present.

A goal conflict based on the shortage of a consumable object can usually be inferred in a story from the following hints:

1. An explicit statement of the shortage of a resource.

2. A statement of the expense of the plans for the character's goals in terms of that resource.

3. A statement of two goals whose plans require the same resource followed by the failure of one of the goals.

For example, story (30) states that John has two goals, and that he has a shortage of a resource. Since the plans for both John's goals require this resource, the reader can infer that John has a goal conflict.

Using the first hint is really a way of invoking Rule C1 above, and assuming that step 3 of that rule holds. Since all these hints may be considered as variations on this rule, I will not bother rewriting the details of the rule for each hint.

The second hint is used in this story:

(31) John wanted to buy a new car and a new house, but they each cost a great deal of money.

Since the story conveys that John plans will consume a lot of money, the reader can assume that John will have some trouble trying to fulfill both goals.

Lastly, in a multi-goal situation in which the plans require the same consumable, fulfilling one goal and failing to fulfill the other indicates a goal conflict. For example, consider

(32) John wanted to buy stereo and an new television set. John bought the television and did without the stereo.

Here John fulfills his goal of owning a television, but fails in his goal of owning a stereo. Since the plans for both these goals require the use of money as a consumable, then the reader can infer that these goals were in conflict based on a limitation of money.

To apply the variations of Rule C1, the reader needs to determine which consumables are common needs of a character's plans. In Chapter 11 I discuss how to determine that a plan uses a consumable object. Briefly, this is done by storing along with each plan a description of the plan's preconditions, and the consequences that result from using that plan. For example, the plan of BARGAIN-OBJECT should be accompanied in memory with the following information:

```
BARGAIN-OBJECT(PLANNER, PERSUADEE, OBJECT, GOAL-OBJECT)

 CONSEQUENCES:

  [ OBJECT IS POSS(PERSUADEE) ]

 CONSTRAINTS:

  [
   OBJECT IS POSS(PLANNER)
   PLANNER IS HUMAN
  ]
```

Since the plan requires the planner to possess the object to be traded, and since possession is transferred as a result of using the plan, the understander can compute that this plan consumes the object traded. Thus to determine if two plans require some of the same consumable objects, the reader needs to access the contraints of each plan, and see if both plans require that the planner possess the same consumable object. Determining that an object is consumed by the plan is discussed in Chapter 11.

The algorithm for determining if a goal conflict exists due to a scarcity of a consumable object is summarized in Figure 7.

### 6.2.3 Mutually Exclusive States

Goals can conflict due to resource limitations because of the nature of the plans chosen for each goal. In addition, goals can conflict because achieving them would require mutually exclusive states to come into existence. For example, consider the following stories:

(33) John wanted to marry Mary, but he also wanted to marry Sue.

(34) John needed a new pair of shoes, but he couldn't bear to part with his money.

(35) John wanted to have a regular job. He also wanted to collect unemployment insurance.

In story (33), the states that would constitute the fulfillment of John's goals are exclusive. A person can be married to only one person at a time, so the achievement of either goal would preclude the achievement of the other. In story (34), the usual plan for getting a pair of shoes is to buy one. A consequence of buying is to change possession of one's money. Since this is inimical to John's other goal of holding on to his money, these goals also conflict due to mutual exclusion. In story (35), a precondition for collecting unemployment benefits is that the person be unemployed. Since this state excludes John having a regular job, the goals in this story are also in conflict.

Figure 7
Algorithm For Detecting Consumption-Based Conflicts

```
 ____________________________         _______________
|                            |       |               |
| Is there a hint that there |  no   | There is no   |
| is a shortage of a         | ====> | conflict based|
| consumable resource?       |       | on a shortage |
|____________________________|       |_______________|

             || yes
             \/
 ____________________________         _______________
|                            |       |               |
| Do the character's plans   |  no   | There is no   |
| each require a common      | ====> | conflict based|
| resource?                  |       | on a shortage |
|____________________________|       |_______________|

             || yes
             \/
 ____________________________         _______________
|                            |       |               |
| Is the amount of the object|  no   | There is no   |
| sufficient for each plan   | ====> | conflict based|
| separately?                |       | on a shortage |
|____________________________|       |_______________|

             || yes
             \/
 ____________________________         _______________
|                            |       |               |
| Is the amount of the object|  yes  | There is no   |
| sufficient for both plans  | =====>| conflict based|
| together?                  |       | on a shortage |
|____________________________|       |_______________|

             || no
             \/
 ____________________________
|                            |
| There is a goal conflict.  |
|____________________________|
```

These examples demonstrate the three ways mutually exclusive states can give rise to goal conflicts:

1. The goals themselves constitute exclusive states. Story (33) was an instance of this situation.

2. One goal state may be excluded by a state that is a consequence of a plan for another goal. A consequence of buying something is not having money, which excluded John's second goal in story (34).

3. A precondition for a plan for one of the goals might exclude the other goal. This was the case in (35), where not having a job was a precondition for one of John's goals, but excluded the other.

The ways in which two goals can exclude each other is summarized in the following rule:

Rule M1:

If

a character has a number of goals such that a goal state, plan consequence, or plan precondition for each goal excludes a goal state, plan consequence, or plan precondition for some other goal,

then

there exists a goal conflict based on mutual exclusion.

To apply rule M1, the reader needs to know how states can exclude one another. There are two ways in which this can happen:

1. The two states are logically exclusive.

2. The two states are socially exclusive.

Logically exclusive states are contradictions, such as being in two different places at once. To determine if two states are logically exclusive, the reader needs to have knowledge about each particular state as well as some general heuristics. For example, a general test for logical exclusion is the following:

Rule M2:

If

one state is the negation of another state,

then

those two states are logically exclusive.

According to Rule M2, if John wanted to be in New York, and wanted not to be in New York, then these two goals are exclusive.

Many special rules are needed here as well. For example, the reader needs to know that a person cannot be in two physically different places at the same time, etc. The full set of such rules is still an open problem.

Socially exclusive states cannot exist at the same time because of cultural prohibitions. For example, John's goals in (33) are socially exclusive because society prohibits a person from being married to two people at once. To determine if two states are socially exclusive, the reader needs to have stored together with each state the social constraints upon that state. For example, for marriage we would have

```
MARRIED(PARTNER1, PARTNER2)
 CONSTRAINTS:
  PARTNER1 IS NOT MARRIED(PERSON1)
  PARTNER2 IS NOT MARRIED(PERSON2)
  PERSON1 IS NOT PARTNER2
  PERSON2 IS NOT PARTNER1
```

When the reader is checking for social exclusion, he compares the constraints of each goal state to the constraints, goal states, preconditions and plan consequences of the other goal looking for logical exclusion. Thus social constraints act like preconditions, except that they impose an arbitrary restriction on the planner. Social exclusion is handled by the rule

Rule M3:

If
  a state has social constraints,
then
  it excludes those states that logically exclude its social constraints.

Being married to someone has the social constraint of not being married to someone else. Since rule M2 states that being married to someone else logically excludes not being married to someone else, being married to someone and being married to someone else are socially exclusive.

In sum, to find out if two goals are mutually exclusive, test their goal states, preconditions, and plan consequences against one another for logical exclusion, and if one of these has social constraints then check those constraints against the rest for logical exclusion.

Note that unlike the resource-based conflicts of the previous section, these conflicts are usually not hinted at by direct references in the story text. The reader must always check for the possibility of an exclusion-based conflict upon learning of two goals of the same characer.

### 6.2.4 Causing a Preservation Goal

Consider the following stories:

(36) John wanted to go to the football game, but it was raining outside.

(37) John wanted to take the night off, but he thought his boss would fire him.

Unlike most of the other stories discussed in this chapter, these stories only have one explicit goal in them. John wants to go to the football game in story (36), and wants to take the day off in story

(37). Nevertheless, these stories constitute goal conflicts. In story (36), if John goes to the football game, then he would become damp, an undesirable state of affairs. In story (37), taking the night off would result in John being fired, another unpleasant state. A character who anticipates being in such a state will have the preservation goal of preventing that state from happening. Thus stories (36) and (37) have goal conflicts between their explicit goals and the unstated preservation goal that would be violated by plans for the explicit goals.

The goal conflict in these stories can be identified by the following rule:

Rule P1:

If
    the plan taken by a character in service of a goal would generate a preservation goal for that character,
then
    a goal conflict exists between the character's actual goal and his anticipated preservation goal.

For example, going to the football game in the rain causes a preservation goal because people normally don't like being rained on; the threat of being fired causes a preservation goal because people normally want to maintain their jobs. Determining that a plan will cause a preservation goal is a matter of determining if the events that constitute that plan will cause something bad to happen to that character. This is essentially the same task that needs to be done to recognize a threat, and requires that the understander have a detailed knowledge of the states people like to avoid, and how these states can come about.

## 6.3 Summary

There are a number of ways in which a character's goals can conflict. The following classification was found useful for detecting goal conflicts:

1. Goal conflicts based on a shortage of resources. These were further broken down into goal conflicts based on a shortage of time, on a shortage of time and functional objects, and on a shortage of consumable objects.

2. Goal conflicts based on mutually exclusive states. These included conflicts based on socially exclusive states and on logically exclusive states. These conflicts also varied according to which states conflicted: The actual goal states, preconditions of plans for those goals, or consequences of performing the plans.

3. Goal conflicts based on generating a preservation goal. These conflicts contained one existent goal, and an anticipated preservation goal that would be generated if the plan for the existent goal were executed.

The next chapter discusses the story situations in which these conflicts arise. This goal conflict classification will be useful in explaining the possible behaviors of a character in each of these goal conflict situations.

# CHAPTER 7

## UNDERSTANDING GOAL CONFLICT RESOLUTIONS

In the previous chapter I introduced the notion of a goal conflict, and described a classification for these conflicts that is useful for recognizing conflicts when they occur. This chapter describes possible story situations in which goal conflicts may arise, and shows how a reader can understand them.

There are three types of situations examined here: goal abandonment, in which a character simply opts for one goal and abandons the others; goal conflict resolution, in which a character tries to resolve the goal conflict and satisfy all his goals; and accidental goal resolution, in which an external event occurs that resolves the goal conflict for the planner.

Most of this chapter is concerned with the resolution of goal conflicts. Understanding goal conflict resolution requires that the reader understand how a character's goals have come to conflict in the first place. Once the reader understands the nature of a goal conflict, he can use knowledge about that type of conflict to explain a character's behavior as part of plan directed at eliminating the conflict.

The classification of goal conflicts used in the previous chapter is used here to organize knowledge about how goal conflicts can be resolved. That is, each of the goal conflicts mentioned previously has a specific set of techniques by which that particular conflict may be resolved. For example, a goal conflict caused by a limitation of time resources can be resolved by extending the deadline by which one of the goals needs to be achieved. A goal conflict based on a limitation of a consumable object can be resolved by acquiring more of the consumable.

## 7.1 Introduction

In the previous chapter the notion of a goal conflict was defined, and algorithms given for recognizing a conflict when it occurs in a story. To this end, a classification for goal conflicts was developed. This classification was found to be useful for organizing the knowledge and inference rules needed to identify a goal conflict.

The reason it was desirable to spot goal conflicts in the first place was to interpret a planner's subsequent actions. That is, when a character has a goal conflict, there are certain courses of action he may take which can be readily explained in terms of his goal conflict. For example, consider the following stories:

(1) John was in a hurry to make an important business meeting. On the way over, he ran into an old girlfriend. She invited him up to her apartment and John accepted. The next day, John's boss told him he was fired.

(2) John was in a hurry to make an important business meeting. On the way over, he ran into an old girlfriend who invited John up to her apartment. John called his boss and asked if the meeting could be postponed.

(3) John had just enough money to buy either a stereo or a television, but he only had enough money for one. John decided to take a second job.

(4) John had just enough money to buy either a stereo or a television, but he only had enough money for one. Then John learned he inherited a small fortune.

John has a goal conflict in each of these stories. In story (2), John simply pursues one of his goals and ignores the other. As a result, John's other goal fails. A reader needs to recognize that this goal conflicts with the goal John fulfilled in order to infer its failure, since the failure of this goal is not explicitly mentioned in the text.

In story (2), John decided to do something about his problem. He tried to get an extension of the deadline for one of his goals, thereby eliminating the conflict. John also tries to eliminate his goal conflict in story (3), this time by acquiring more of a consumable object. In both stories, John tries to do away with one of the causes of the conflict, but in each case his actions are different because they address a different class of goal conflicts.

John's goal conflict disappears in story (4), but not through any action by John. Here another person undid one of the factors contributing to the goal conflict. A reader of this story needs to understand that this fortuitous circumstance freed John of his problem in order to understand that John could now fulfill both goals.

### 7.1.1 Goal Conflict Situations

These stories are examples of a number of different situations that might take place after the occurrence of a goal conflict. These stories fall into three kinds of goal conflict situations:

1. Goal Abandonment

   The character can simply opt for one of the conflicting goals and try to fulfill it, abandoning the other goals. For example, story (1) above is an instance of goal abandonment. John spent his time with his old girlfriend and as a result he failed to make it to his meeting on time.

2. Goal Conflict Resolution

   The character can try to resolve the conflict. If he succeeds, then he can try to fulfill all his goals. If he fails to resolve the conflict, then he can still opt for one goal via goal abandonment. Stories (2) and (3) are attempts at goal conflict resolution, because the character whose goals are in conflict tries to undo one of the factors that caused the goals to conflict.

3. Spontaneous Goal Conflict Resolution

   The conflict may be resolved by some event by a character other than the planner or his agent. For example, in (4), John's inheritance eliminated one of the causes of John's goal conflict, thus making that conflict evaporate.

Each of these options constitutes a different goal conflict situation. That is, each situation organizes knowledge about how to understand an event following the statement of a goal conflict. Thus an understander must be capable of understanding which goal conflict situation is occurring in a story in order to explain the behavior of the planner. Each of these goal conflict situations are now considered in turn.

## 7.2 Abandoning a Goal

This is the simplest goal conflict situation. The character with the conflict merely pursues one of the goals, and the reader infers that the other goal is abandoned. For example, consider the following story:

(5) John wanted to watch the Monday night football game. He also had a paper due the next day. That night, John watched the football game. John failed Civics.

In story (5), John pursued his goal of watching the football game, and the reader infers that his other goal is abandoned. This requires the following rule:

Rule S1:

If

1. A planner has a goal conflict,

and

2. The planner tries to fulfill one of the conflicting goals,

then

The goals conflicting with this goal will fail.

Rule S1 is used in story (5) to infer that John didn't write his paper because he watched the football game. This inference can then be used to explain why John failed the course: Handing in an assignment is instrumental to passing a course. If a goal instrumental to another goal fails to be achieved, then the goal to which it is instrumental also fails.

## 7.3 Resolving a Goal Conflict

Sometimes a goal conflict can be resolved. For example, consider the following stories:

(6) John wanted to watch the football game but he had a paper due the next day. John called his teacher and asked if he could have an extension.

(7) John wanted to watch the football game but he had a paper due the next day. John watched the game. Then he stayed up all night to write his paper.

(8) John wanted to watch the football game but he had a paper due the next day. John decided to record the football game on his Betamax.

(9) John wanted to cook linguine and hamburgers on his camp stove at the same time, but it only had one burner. John walked over to a neighboring camper and asked him if he could borrow another stove.

(10) John was camping up in the hills when he discovered he didn't have enough water to wash both his hands and his dishes. John decided to look for a stream.

(11) John wanted to go to the football game, but it was raining outside. John decided he would go to the game and carry a huge umbrella.

Each of these stories expresses a goal conflict followed by an attempt to resolve it. The way in which each conflict is resolved depends largely on the type of goal conflict. For example, in stories (6), (7), and (8) John has a goal conflict based on a shortage of time resources. He tries to resolve it by eliminating the deadline on his paper in story (6). In story (7), John achieves both goals by abandoning an implicit goal of getting some sleep, and in story (8) uses a different plan to fulfill one goal that does not impose the same time restrictions as the original plan.

In story (9), John's conflict is based on the inadequacy of a functional item. It is remedied by the acquisition of another functional item. Story (10) has a shortage of a consumable, which is addressed by a plan to obtain more of that consumable. Finally, story (11) contains a goal conflict based on invoking a preservation theme, and is resolved by a plan to counteract the anticipated threat.

Thus the way in which a character can try to resolve a goal conflict is a function of the type of goal conflict. In this section, which constitutes the bulk of this chapter, I consider the various types of goal conflicts characterized in the previous chapter and examine the plans by which they may be alleviated. Recall that the following classes of goal conflicts were presented:

1. Goal conflicts based on a shortage of resources. These were further broken down into goal conflicts based on a shortage of time and on a shortage of consumable objects.

2. Goal conflicts based on mutually exclusive states. These included conflicts based on socially exclusive states and on logically exclusive states. These conflicts also varied according to which states conflicted: The actual goal states, preconditions of plans for those goals, or consequences of performing the plans.

3. Goal conflicts based on invoking the preservation theme. These conflicts contained one existent goal, and an anticipated preservation goal that would be generated if the plan for the existent goal were executed.

### 7.3.1 Resolving Resource Shortages

Goal conflicts that are based on a limitation of some resource can be resolved in one of two ways:

1. The planner can obtain more of the scarce resource.

2. The planner can alter the plans for his goals to eliminate the need for some of the resource.

Acquiring more of the scarce resource is dependent upon what that resource is. Thus the rules for eliminating resource shortage are a function of the resource distinctions made in the previous section.

For example, consider time-based resource shortages. In a goal conflict based on a shortage of time, the following conditions are true:

1. The total time required by the character's plans sequentially was greater than that available to the character.

2. The plans could not be executed simultaneously.

For example, in a story like (5) above, John has only a certain period of time within which to write his paper and watch the football game, and this time period is insufficient to execute one plan after another. Also, John could not execute both plans simultaneously, because watching a football game would take too much of his attention away from writing the paper.

To eliminate this kind of goal conflict, a character must find some more time, or find a way to execute the plans for each goal simultaneously. There are a number of conditions that must be true in order for the conflict to exist. A character can eliminate a conflict by undoing one of these conditions. To understand a situation in which a character tries to undo a state to resolve a goal conflict, a reader must know what the states are that contribute to that conflict.

7.3.1.1 Factors involved in time-based goal conflicts

In order for there to be insufficient time to execute the plans sequentially, the following conditions must hold:

1. The plans must have a time restriction upon them.

2. The time required to execute the plans must be greater than the time available to the planner.

The plans must be restricted in initiation or else there will always be enough time to do them sequentially. The time required must be greater than the time available or else the planner could simply do the plans for the goals one after the other.

A plan might be restricted in initiation for one of two reasons:

1. The plan may require synchronization with some event not controlled by the planner.

2. The goal itself may contain a deadline by which it must be achieved.

For example, for John to watch the football game in story (5) he had to synchronize his watching the game with the time during which the game was televised. Other examples of plans that require synchronization are going to a store while it is open, going to see a person while that person is home, and getting on a bus when the bus arrives.

To compute the time interval for a plan requiring synchronization, the following rule is used:

Rule S2:

If

a plan involves an event whose actor is not the planner, nor an agent employed by the planner,

then

the plan must begin during the time that the event occurs.

Often, as is the case with watching the football game, there is only one point in time at which a successful plan can be initiated. That is, it is not possible to start earlier and finish it earlier, because the start of the game is a fixed point; the actual act of ATTENDing to the game is essentially frozen in time.

An example of a goal that contains a deadline was the second goal of story (7), where the goal required the paper be ready the next day. Other examples of goals that contain deadlines include the instrumental goal of having a bet placed with a bookie before the start of a race, and being out of a burning house before being reached by the flames. In all these cases, the interval of initiation contains all the points that occur before the deadline by a time equal to the duration of the plan. That is, we have the following rule:

Rule S3:

If

part of a goal is an explicit predication of a time by which that goal must be achieved,

then

a plan for that goal must be initiated by a time not less than the deadline minus the duration of the plan.

In addition to being restricted in time, the time required by the plans must be greater than the time available. The time available to a character is time not already scheduled for "background" goals. A background goal is a recurring goal whose plan has a fixed place in most character's schedules. For example, the most usual background goals are Satisfy-hunger and Satisfy-sleep.

The notion of available time is important because time-based goal conflicts are usually stated in terms of available time rather than absolute time. That is, the fulfillment of Satisfaction goals like Satisfy-sleep and Satisfy-hunger normally require a fixed allotment of each individual's time. When a story refers to a time-based goal conflict, it is assumed that the planner is trying to fit these goals into his available time, so that a goal conflict may exist even if there is enough time in the absolute sense to execute both plans.

The second condition that must hold if a time-based goal conflict is present is that the planner cannot do all the plans at once. Two plans may exclude each other from being executed at the same time for one of the following reasons:

1. The plans each require the use of a functional object, and the number of functional objects available to the planner is insufficient.

2. The plans together would exceed the planner's abilities. That is, two goals cannot be executed at the same time if this would require more of a person's physical or mental resources than can be delivered simultaneously.

3. One plan has a precondition which violates some state necessary for the success of another plan.

For example, consider story (12)

(12) John wanted to fill up his waterbed and fill up his swimming pool, but he only had one hose.

The two plans in (12) each require the use of a hose, so the goals for each of these plans are in conflict. This is an instance of the following rule:

Rule S4:

If

a character has a set of goals whose plans each requires the use of the same kind of functional object,

then

that character can execute simultaneously no more plans than the number of those functional objects that he has control over.

Since there are two goals in story (12), each of whose plan requires a hose, and since John has control of only one hose, then according to rule S4, he cannot execute the plans simultaneously.

A set of plans that cannot be executed simultaneously because of the capabilities of the planner exist in story (7). Watching a football game requires a portion of a person's attention, and writing a paper demands all of a person's attention. Together the two plans require

more attention than a person has available. In constrast, there is no goal conflict in the story

(13) John wanted to watch the football game, but he also wanted to eat popcorn.

The plans for these goals could be simultaneous because eating presumably does not require much attention. Thus we have the following rule:

Rule S5:

If

a planner is limited in of one of his abilities,

then

he can do only as many plans that require a portion of one of his abilities as that ability allows.

That is, associated with each plan, the understander must have stored the plan's demands on the planner's abilities. I currently distinguish the following types of abilities: attention, intelligence, strength, and speed. For example, writing a paper normally requires a person's full attention, so it cannot be done while planner is attempting to execute another plan.

The third way in which two plans may exclude each other from occurring simultaneously is when one plan has some precondition which would contradict some state necessary for another plan to succeed. For example, compare the following stories:

(14) John wanted to take the job in Rochester. He also wanted to live in Berkeley.

(15) John wanted to take the job in Berkeley. He also wanted to live in Berkeley.

In story (14), taking the job in Rochester would require living near Rochester, and this would exclude the state of living in Berkeley, John's other goal. Thus the two states cannot both be maintained simultaneously. In contrast, since living in Berkeley is compatible with taking a job in Berkeley, there is no goal conflict in story (15). This case was an instance of the rule:

Rule S6:

If

the execution of two plans requires two mutually exclusive states to exist,

then

those plans cannot be executed simultaneously.

The notion of mutually exclusive states is important here in recognizing that two plans could not occur simultaneously. The notion of mutual exclusion with respect to goal conflicts was discussed further in the previous chapter.

In sum, if two plans cannot be executed simultaneously, then one of the following conditions must hold: The plans require the use of a functional object which is in short supply, the plans require more of some ability than the planner is capable of, or the plans require the existence of mutually exclusive states.

The point of going into all this detail about time-based goal conflicts is that an attempt by a character to change any of these factors is a plan to resolve the goal conflict. For example, a planner can try to resolve a time-based goal conflict by trying to buy more time for his plans. This can be accomplished in the following ways:

1. The planner can try to eliminate the time restrictions that determine the time by which his plans must be accomplished.

2. The character can forego satisfaction of a background goal.

For example, story (6) above is an instance of a goal conflict based on a time shortage that the planner addresses by trying to change a deadline:

(6) John wanted to watch the football game but he had a paper due the next day. John called his teacher and asked if he could have an extension.

In this story, the time restrictions on writing the paper are imposed by another actor. John can therefore try to change these restrictions by persuading the actor imposing them to change them.

A restriction is imposed by another actor if the planner must synchronize his plan with an event performed by another actor, or if the goal state has a deadline set by another actor. For example, in story (6), the deadline on writing the paper is presumably imposed by the teacher. Also, the time of the football game is set by someone other than John. Thus John can eliminate the conflict if he persuades his teacher to change the deadline, or persuades the producers of the football game to air it at a different time.

This goal conflict resolution strategy can be summarized by the following rules:

Rule S7:

If

1. a goal conflict is due to a shortage of time,

and

2. one of the time restrictions is due to another actor,

then

the planner can resolve the conflict by persuading that actor to change the restriction.

Besides persuading another actor, the planner can forgo satisfying a background goal. Recall that a background goal is a recurring goal, such as eating and sleeping, that normally occupies part of every character's day. Since it is possible to postpone the fulfillment of one of these goals for a while, an actor can reclaim some time by not pursuing one of these. This occurred in example (7) above:

(7) John wanted to watch the football game but he had a paper due the next day. John watched the game. Then he stayed up all night to write his paper.

Here John gets back some more time by abandoning his Satisfy-sleep goal. In effect, John changed his goal conflict to a conflict between writing a paper and getting some sleep, and then abandons the latter goal. This is summarized in the following rule:

Rule S8:

If

a goal conflict is due to a shortage of time,

then

the planner can resolve the conflict by abandoning a background goal whose plan normally consumes at least as much time as needed to allow the completion of the conflicting goals.

In addition to finding more time, the planner can resolve a conflict by changing his plans so that the shortage of time is eliminated. For example, consider the following stories:

(8) John wanted to watch the football game but he had a paper due the next day. John decided to record the football game on his Betamax.

(16) John wanted to watch the football game but he had a paper due the next day. John decided to hire someone to write the paper for him.

In story (8), John alters his plans to use a new plan that does not require him to synchronize his plans with the broadcast of the game. In story (16), John uses a plan that gives John more time by getting someone else to do his work for him. This planning can be summarize in the following rules:

Rule S9:

If
a goal conflict is due to a shortage of time,

then
the planner can resolve the conflict by changing to a plan that either requires less time or has a more lenient time restriction.

Rule S10:

If
a goal conflict is due to a shortage of time,

then
the planner can resolve the conflict by employing an agent to achieve one of the goals.

One additional technique for eliminating time-based goal conflicts is applicable when the time shortage involves the use of a functional object. For example, recall story (9) above:

(9) John wanted to cook linguine and hamburgers on his camp stove at the same time, but it only had one burner. John walked over to a neighboring camper and asked him if he could borrow another stove.

Here John needs more functional capacity within the same amount of time. He tries to achieve it by acquiring more functional objects. Thus we have the following additional rule:

Rule S11:

If

a goal conflict is due to a shortage of time and to the inadequacy of a functional object,

then

the planner can resolve the conflict by acquiring an additional functional object, or by acquiring a functional object of greater capacity.

Resource shortages are caused by shortages of consumable objects in addition to time shortages. This was the case in example (3):

(3) John had just enough money to buy either a stereo or a television, but he only had enough money for one. John decided to take a second job.

This example contains a strategy similar to that used in the above example. The planner tries to eliminate the conflict by acquiring more of the short resource. Story (3) is an instance of the rule

Rule S12:

If

a goal conflict is due to a shortage of a consumable object,

then

the planner can resolve the conflict by acquiring more of that consumable.

Consumable object shortages are also subject to plan changing. For example, consider:

(17) John had just enough money to buy either a stereo or a television, but he only had enough money for one. John bought a stereo and stole a television set.

John has changed from a plan involving the consumable money to a plan that does not, in accordance with the rule

Rule S13:

If

a goal conflict is due to a shortage of a consumable object,

then

the planner can resolve the conflict by changing to a plan that requires less of that consumable substance.

If a reader of story (17) were asked the question

Q) Why did John decide to steal the television set?

The inference made using Rule S13 provides the answer:

A) Because John no longer had enough money left to buy one.

7.3.1.2 Computer Example

The following is a trace of a PAM run on a story involving a time-based goal conflict:

[PHOTO: Recording initiated Fri 11-Aug-78 4:35PM]

@RU PAM

*(UNDERSTAND CD11)

THE STORY IS

JOHN WANTED TO WATCH THE FOOTBALL GAME,
BUT HE HAD A PAPER DUE THE NEXT DAY.
JOHN WATCHED THE FOOTBALL GAME.
JOHN FAILED CIVICS.

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(JOHN WANTED TO WATCH THE FOOTBALL GAME) | |
| CONCEPTUALIZATION IS:<br>((CON<br>((ACTOR HUMO <=> (*ATTEND*)<br>OBJECT PARTO TO ACTIVO)<br>TIME (FORM6))<br>IS (*GOAL* PART HUMO))<br>TIME (FORM7)) | ACTIVO is the football game that HUMO (John) wants to watch. |
| NOT A PREDICTED INPUT | |
| BEGIN SEARCH FOR EXPLANATION | |

```
TESTING EXPLANATION OFFERED BY
EENTERTAINMENT-REQ

EXPLANATION IS GOAL:
(*EENT* PLANNER HUMO ACTIVITY
ACTIVO)
 WITH SOURCE (*ENJOYMENT*)

EXPLANATION CONFIRMS PREDICTION
INIT-REQ

FOUND EXPLANATION SEQUENCE:

EENTERTAINMENT-REQ
```

PAM finds EENTERTAINMENT-REQ, which states that watching football is usually an "enjoy entertainment goal" (*EENT*). PAM adds this inference to the story representation.

```
*** ADDING TO STORY REPRESENTATION:

 GOAL: (*EENT* PLANNER HUMO
ACTIVITY ACTIVO)

LOADING PREDICTION
EENT-ATTEMPT-REQ
```

A request is loaded looking for an attempt at fulfilling this goal.

```
NEXT INPUT IS:
 (BUT HE HAD A PAPER DUE THE NEXT
  DAY)

CONCEPTUALIZATION IS:
((CON
  ((ACTOR PHYSO IS (*EXIST*))
   TIME (FORM12))
  IS (*GOAL* PART HUMO))
 TIME (FORM13))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION
OFFERED BY AASSIGNMENT-REQ

EXPLANATION IS GOAL:
(*AASS* PLANNER HUMO ASSIGNMENT
PHYSO DUE (TIMK4))
```

Having a paper due means that John has to complete an assignment for a course. Thus PAM finds request AASSIGNMENT-REQ which infers that John has the goal of completing an assignment by a specific time. An explanation for this goal is now sought.

```
NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
```

CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
AGRADE-REQ

EXPLANATION IS GOAL:
(*AGRD* PLANNER HUMO COURSE (NIL))
PLAN: ($DOCOURSEWORK PLANNER HUMO
COURSE (NIL))

WITH SOURCE (*SOCIALROLE* ROLE
($STUDENT)

Doing an assignment is usually instrumental to achieving a good grade (*AGRD*) by doing coursework Furthermore, this goal come from the social role of being a student. Since being in a social role is a theme, an explanation has been found.

EXPLANATION CONFIRMS PREDICTION
INIT-REQ

FOUND EXPLANATION SEQUENCE:

AASSIGNMENT-REQ -> AGRADE-REQ

*** ADDING TO STORY REPRESENTATION:

These goals are added to the representation.

***** GOAL CONFLICT DETECTED

CONFLICT IS BETWEEN
GOAL: (*AASS* PLANNER HUMO
ASSIGNMENT PHYSO DUE (TIMK4))
AND
GOAL: (*EENT* PLANNER HUMO
ACTIVITY ACTIVO)

***** CONFLICT IS BASED ON SHORTAGE
OF TIME RESOURCES

Since the goal of doing the assignment (*AASS*) has an explicit due date, this is a hint at a time-based goal conflict. PAM checks for other goals for the same character, and find the "enjoy entertainment" goal (*EENT*) inferred from the last input. Thus PAM infers a goal conflict based on a shortage of time.

LOADING PREDICTION
ABANDON-GOAL-REQ

LOADING PREDICTION
CHANGE-DEADLINE-REQ

LOADING PREDICTION
ABANDON-BACKGRD-GOAL-REQ

A set of rules is now loaded to handle various situations that may follow. For example, ABANDON-GOAL-REQ looks for one goal to be pursued and the other abandoned; CHANGE-DEADLINE-REQ looks for an attempt to buy more time; and ABANDON-BACKGRD-GOAL-REQ looks for the abandonment of a background goal.

INFERRED GOAL: (*AASS* PLANNER HUMO
ASSIGNMENT PHYSO DUE (TIMK4))

```
INFERRED GOAL: (*AGRD* PLANNER HUMO
COURSE (NIL))


NEXT INPUT IS:
 (JOHN WATCHED THE FOOTBALL GAME)

CONCEPTUALIZATION IS:
((ACTOR HUMO <=> (*ATTEND*) OBJECT
  PARTO TO ACTIVO)
 TIME (FORM26))


NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
WATCH-PLAN-REQ
```

PAM infers that this is part of a plan to enjoy the football game.

```
EXPLANATION IS
 PLAN: (*PB-WATCH* PLANNER HUMO
ACTIVITY ACTIVO)

EXPLANATION CONFIRMS PREDICTION
EENT-ATTEMPT-REQ
```

PAM infers that the input is a plan to fulfill the first goal.

```
FOUND EXPLANATION SEQUENCE:

WATCH-PLAN-REQ


*** ADDING TO STORY REPRESENTATION:

INPUT CONFIRMS PREDICTION
EENT-ATTEMPT-REQ

 OUTCOME OF GOAL: (*EENT* PLANNER
HUMO ACTIVITY ACTIVO)
 IS (*SUCCEED*)


INPUT CONFIRMS PREDICTION
ABANDON-GOAL-REQ

PREDICTING FAILURE FOR GOAL:
(*AASS* PLANNER HUMO ASSIGNMENT
PHYSO DUE (TIMK4))

BASED ON GOAL CONFLICT
```

**One of the rules previously loaded predicts the failure of one goal if a plan for the other is executed. This rule now predicts the failure of one of John's goals, and requests are loaded that will make this inference upon encountering certain events.**

```
LOADING PREDICTION
PROP-FAILURE-REQ

LOADING PREDICTION
FAILURE-REQ


NEXT INPUT IS:
 (JOHN FAILED CIVICS)

CONCEPTUALIZATION IS:
((ACTOR HUMO <=>
  ($TAKECOURSE STUDENT HUMO
   COURSE (*CIVICS*) OUTCOME
   (*FAILURE*)))
 TIME (FORM34))


INPUT CONFIRMS PREDICTION
PROP-FAILURE-REQ

SUBGOAL FAILURE PROPAGATION
INFERRING OUTCOME OF GOAL:
(*AASS* PLANNER HUMO ASSIGNMENT
PHYSO DUE (TIMK4)) IS (*FAILURE*)

 OUTCOME OF GOAL: (*AGRD* PLANNER
HUMO COURSE (NIL)) IS (*FAILURE*)


FINISHED UNDERSTANDING PHASE
```

One of the rules loaded to detect goal failure for a particular goal works this way: If it sees failure for a goal to which the particular goal is instrumental, then it infers failure for the particular goal. This rule is used here to infer that John failed to write his paper.

---

```
QUESTION: ?Q1

 Why did John fail a course in Civics?
 He failed to hand in an assignment.

[PHOTO:  Recording terminated  Fri 11-Aug-78 4:36PM]
```

To answer the last question, PAM tries to find a reason why this event occurred. The reason for this event is represented as the failure of an instrumental goal. PAM expresses the failure of this goal as the answer to the question.

### 7.3.2 Resolving mutually exclusive states

A goal conflict can also arise through mutually exclusive states. The ways in which these conflicts can be resolved depend upon the way in which the states exclude one another. For example, consider the following stories:

(18) John wanted to marry Mary. He also wanted to marry Sue.
He decided to marry both of them.

The goal conflict in story (18) is based on socially exclusive states. John can therefore try to resolve the conflict by violating the social rule that caused the conflict in the first place. This resolution is not complete, however. Violating social rules usually has undesirable consequences. Since people usually want to prevent undesirable events from befalling them, this attempt at a resolution will produce a new goal conflict based on invoking the preservation theme. For example, John's decision to marry both women in story (18) might cause his society to arrest him for bigamy, or subject him to social ridicule.

Thus the resolution of goal conflicts based on socially exclusive states can be done according to the following rule:

Rule S14:

If

a goal conflict is due to socially exclusive states,

then

the planner can resolve the conflict by violating the social rule causing the conflict, and incurring a new goal conflict between his old goals and the preservation goals that would arise from the social rule being violated.

One other way to eliminate a socially based goal conflict is to change one's society. For example, consider the following story:

(19) John wanted to marry Mary. He also wanted to marry Sue.
He decided to become a Muslim and move to Saudi Arabia.

Islam's social rules permit multiple marriages. Thus by becoming part of this society, the social rule underlying John's goal conflict disappears. Thus we have the rule

Rule S15:

If

a goal conflict is due to socially exclusive states,

then

the planner can resolve the conflict by becoming part of a society in which the desired states are not socially exclusive.

At first it might seem that goal conflicts based on logically exclusive states are not resolvable. However, this is not always the case. For example, consider

(20) John wanted live in San Francisco, but he also wanted to live near Mary, and she lived in New York. John tried to persuade Mary to move to San Francisco with him.

The goal conflict in story (20) is based on the logically exclusive states of John living near New York, and John living near San Francisco. However, John could still eliminate the conflict if he could persuade Mary to move with him. This is possible because John's goal of living near Mary does not automatically preclude his living in San Francisco; the exclusion comes about through an inference that living near Mary implies living near where she lives. That is, the logically exclusive states in story (20) are not inherent in John's goal, but are a product of the plans he has chosen for them. John can plan around this conflict by changing a condition upon which this inference is based. Thus we have the following technique for eliminating logically exclusive states:

Rule S16:

If

1. a goal conflict is due to mutually exclusive logical states,

but

2. the exclusion depends upon a third state,

then

the conflict can be eliminated by changing the third state so that the inference causing the goal conflict is no longer valid.

The third state in story (20) was Mary's living in New York. Since this state was used in the computation that determined John's goals to be exclusive, John can remedy the situation by changing this state so that John being in San Francisco would not exclude living near Mary. That is, John could try to change this state so that Mary would live near San Francisco.

In processing story (20), when the reader sees John trying to persuade Mary to live in San Francisco, the reader will infer that John wanted Mary to live in San Francisco. Since the reader knows that John had a goal conflict and may be trying to resolve it, the reader checks to see if this could be an attempt at a resolution. Since the goal conflict was based on logically exclusive states, Rule S16 tells the reader to check to see if John is trying to change a state causing the exclusion relationship.

This is done by trying to compute if the goal conflict would still exist if this state were achieved. Since no goal conflict would be present if Mary lived in San Francisco, the reader can explain John's behavior as part of a plan to eliminate the basis for the incompatibility of two states.

Goal conflicts based on exclusion can also be eliminated if one of the states causing the conflict is a consequence or precondition of a plan for one of the goals. For example, consider

(21) John needed a new pair of shoes, but he couldn't bear to part with his money. He decided to steal a pair.

In story (21), the goal conflict arose because buying the shoes has the consequence of John transferring his money to someone. John decided to eliminate this conflict by using a plan in which this consequence is avoided, namely, stealing the shoes. In general,

Rule S17:

If

1. a goal conflict is due to mutually exclusive states,

and

2. one of those states is a precondition or consequence of a plan for one of the goals,

then

the conflict can be resolved by changing to a plan that does not have the undesirable precondition or consequence.

### 7.3.3 Resolving preservation based conflicts

The fourth way in which goals can conflict occurs when the fulfillment of one goal would invoke the preservation theme. These goal conflicts can be resolved by changing plans, or by planning for the preservation goal. For example, consider the following stories:

(22) John wanted to see the football game, but it was raining outside. John decided to stay home and watch the game on television.

(23) John wanted to see the football game, but it was raining outside. John decided he would go to the game and carry a huge umbrella.

John has a goal conflict in both these stories because going to a football game in the rain may cause one to get wet, and this results in an undesirable state. In story (22), John eliminates the conflict by changing the plan by which he was going to fulfill his goal of watching the football game. Since the plan for watching the game on television does not result in the character getting wet, the preservation theme would not be invoked, and the goal conflict is avoided. Thus we have the following rule:

Rule S18:

If

a goal conflict is caused by invoking the preservation theme,

then

the conflict can be resolved by changing to a plan that does not invoke that theme.

Another way to resolve a goal conflict caused by generating a preservation goal occurs in story (23). Here John plans for the preservation theme in advance, so that when he executes the plan for his original goal, his preservation goal is also fulfilled. That is, using an umbrella is a way to keep oneself from getting wet in the rain. When the reader learns that John decided to carry the umbrella, the reader infers that John wanted to use the umbrella to keep dry. Since being dry is the preservation goal that arises if John goes to the football game, the reader can infer that John's getting the umbrella was part of a plan to deal with the preservation goal and still achieve the original goal. This is an instance of the rule:

Rule S19:

If

a goal conflict is caused by invoking the preservation theme,

then

the conflict can be resolved by planning for the preservation goal in advance.

Another example of this rule occurs in the following story:

(24) John wanted to rob a bank, but he was afraid the police would find out he did it and arrest him. He decided to wear a mask.

John has a goal conflict between the goal of having some money and the goal of preserving his freedom. Since John anticipates this goal, he can plan for it in advance. In order to arrest John, the police have to find out that he committed the crime. John is therefore trying to prevent the police from finding out that he robbed the bank by making it hard for people to recognize him. Thus John is trying to resolve his goal conflict by planning for a preservation goal in anticipation of its occurrence.

### 7.3.4 Resolving by goal substitution

There is one more strategy for resolving goal conflicts that is applicable to any situation in which a character may fail to achieve a goal. For example, consider the following stories:

(25) John wanted to see the football game, but his wife said she would divorce him if he watched one more game. John settled for watching the tennis match together with his wife.

(26) John wanted to live in San Francisco, but he also wanted to live near Mary, and she lived in New York. John decided he could probably find another girlfriend in San Francisco.

The strategy involved in these stories is goal substitution. In story (25), John wanted to watch the football game, but this created a goal conflict because it would invoke the preservation theme. John resolves this conflict by changing his goal from watching the football game to watching the tennis match. In story (26), the goal conflict is between being near a girlfriend or living in San Francisco. John resolves this conflict by changing his goal from being near Mary to being near some girlfriend.

Stories (25) and (26) are instances of the following rule:

Rule S20:

If

a character has a goal conflict,

then

the goal conflict can be resolved by substituting a new goal that does not conflict with the other goals for one of the conflicting goals.

Rule S20 is part of a more general strategy concerning goal failure: If a planner anticipates failing to achieve a goal, he can substitute another goal that may be easier to achieve.

### 7.4 Having the Conflict Resolved For You.

The stories in the previous section were all cases in which the character having the goal conflict took some action to resolve it. However, a goal conflict can be resolved by the action of a character other than the planner. For example, consider the following stories:

(27) John wanted to watch the football game, but he had a paper due the next day. Then the teacher postponed the assignment for a week.

(28) John enjoyed smoking marijuana, but he was paranoid about getting busted. Then one day, the city council voted to legalize drugs.

The resolutions to these conflicts follow essentially the same rules as they did in the last section. For example, changing the due date of the assignment in story (27) resolves the conflict in that story by changing a deadline. The city council's legalization of marijuana in story (28) resolves that conflict by removing one of the factors that was causing John to have preservation goals.

When a reader encounters an event following the statement of a goal conflict, the reader must check to see if that event resolves the conflict, even if the actor of that event is a different actor than the one who has the conflict. The process of determining that the conflict has been resolved is not substantially changed. For example, Rule S7 above was as follows:

Rule S7:

If

1. a goal conflict is due to a shortage of time,

and

2. one of the time restrictions is due to another actor,

then

the planner can resolve the conflict by persuading that actor to change the restriction.

To handle the case where another actor resolves the goal for the planner, this rule would have to be divided into two parts:

Rule S7′:

If

1. a goal conflict is due to a shortage of time,

and

2. one of the time restrictions is due to another actor,

then

the conflict can be resolved if the time restriction changes.

Rule Control:

If

1. some condition contributes to a goal conflict for a character,

and

2. some other character has control over that condition,

then

the first character may try to convince the second to change the condition.

Rule Control is a general rule about persuading a person who can ease the conflict. Rule S7′ states a way in which a condition can contribute to a conflict. The other rules of the previous section can be similarly modified to produce rules like S7′ stating conditions that contribute to conflicts. These new rules will handle the situations in which the goal conflict is resolved for the planner, and together with Rule Control, they explain the behavior of a character trying to persuade someone in power to resolve the conflict.

## 7.5 Summary

There are a number of story situations based upon goal conflicts. A goal conflict may cause a character to fail to achieve a goal; it may cause a character to try to resolve the conflict; or the conflict may be resolved by an incidental action. To understand these situations, the reader must be able to detect goal conflicts, and use knowledge about the nature of goal conflicts to explain the behavior of a story's characters.

Explaining the behavior of a character experiencing a goal conflict is a function of the type of goal conflict. Each type of goal conflict presupposes a set of conditions that made the goals conflict. A planner could try to resolve the conflict by undoing one of these conditions. If a reader possesses knowledge about each type of goal conflict situation, then the reader can recognize which actions are instances of plans to resolve a goal conflict, thus explaining the character's behavior.

# CHAPTER 8

## GOAL COMPETITION

Stories usually have more than one character. When a number of characters' goals are active at the same time, these goals may interact to give rise to interesting story situations. One such class of story situations is goal competition. Goal competition arises when several characters' goals are active, and the fulfillment of one character's goal will interfere with the fulfillment of the goals of the other characters.

Goals can compete with one another for reasons similar to those underlying goal conflict. A story understander needs to know these reasons in order to identify goal competition when it occurs in a story, and to explain the behavior of characters whose goals compete. The classification of types of goal competition needed for these purposes is almost identical to that used for goal conflict detection. There is goal competition based on the limitation of a shared resource, goal competition caused by mutually exclusive states among the characters' plans or goals, and goal competition that occurs when the fulfillment of one character's goal will cause another character to have a preservation goal.

However, there are some significant differences between the conditions for goal conflict and for goal competition. For example, recognizing that two people's goals compete due to a limitation of resources requires the reader to determine that the resource is shared between them. The notion of a shared resource did not arise in the study of goal conflicts, where the goals in question all belonged to the same character.

This chapter shows how this classification system can be used to organize the knowledge needed to detect goal competition. The next chapter deals with understanding the situations in which goal competition arises. The differences between goal competition and goal conflict manifest themselves most significantly in the next chapter. Although the bases for the each kind of situation are similar, the situations themselves are quite distinct. For example, a character who

recognizes that his goals compete with those of another character may take some action to thwart that character. That character may in turn try to stop his competitor from foiling his plans. Thus goal competition often gives rise to complex interactions among story characters that have not been dealt with so far.

## 8.1 Introduction

In the previous chapter I showed how a character's goals could conflict with one another and thereby create difficulties for that character. A person's goals can also be adversely affected by goals possessed by other characters. In these situations, the reader cannot follow each character's goals independently. The reader must understand the connection between the different character's goals in order to explain each character's behavior.

For example, consider the following stories:

(1) John told Mary he wanted to watch the football game. Mary said that she wanted to watch the Bolshoi ballet. Mary put on channel 3. John got out the lawnmower.

(2) John told Mary he wanted to watch the football game. Mary said she wanted to bake cookies. John turned on the television while Mary mixed some batter.

(3) John wanted to win the high hurdles. Bill also wanted to win the high hurdles. John won the race. Bill was very upset.

(4) John wanted Mary to win the high hurdles. Bill also wanted Mary to win the high hurdles. Mary won the race. John and Bill were very happy.

Each of these stories has two characters that have goals. In stories (2) and (4) the goals of each character are not in an adverse relation to one another, and the reader can understand each character's action by following each goal independently. In fact, the events bearing on each goal can be separated into two comprehensible stories. For example, story (2) could be made into the following two stories:

(5) John said he wanted to watch the football game. John turned on the television.

(6) Mary said she wanted to bake cookies. Mary mixed some batter.

In each of these stories, the explanation found for each character's action is the same as the explanation found in story (2). John's turning on the television was instrumental to his watching the football

game in both story (5) and story (2); Mary's mixing batter was part of a plan to make cookies in stories (6) and (2).

Similarly, story (4) could be separated into two independent stories:

(7) John wanted Mary to win the high hurdles. Mary won the race. John was very happy.

(8) Bill wanted Mary to win the high hurdles. Mary won the race. Bill was very happy.

The explanation for Bill's and John's happiness in these stories in the same explanation that a reader would find in story (4): When something a person wants comes to pass, then that person will often become happy. This explanation would be found in stories (7), (8) and (4) by considering each goal independently.

Stories (1) and (3), on the other hand, cannot be understood without realizing that the characters' goals are related. Consider what happens if we remove one of the character's goals from these stories. For example story (1) would become:

(9) John said he wanted to watch the football game. Mary put on channel 3. John got out the lawnmower.

and story (3):

(10) John wanted to win the high hurdles. John won the race. Bill was very upset.

Most readers of story (9) experience some difficulty finding an explanation for John's action. Since Mary's goals are unknown, the reader might assume that Mary was turning on the television for John. Then when John's action suggests that he is not going to watch the game, the reader must find a new explanation for John's action.

In story (1), the explanation for this action was readily found: Mary's goal was in opposition to John's. Her action in service of her goal therefore excluded John from fulfilling his. John decided to pursue some other goal instead, and got out the lawnmower as part of a plan for this goal. The reader could make this inference in story (1) because the reader knows Mary's goal in that story, and can infer the relationship between Mary's goal and John's.

Likewise, a reader of (10) probably has no ready explanation to explain Bill's becoming upset. But in story (3) the explanation is clear: John's winning the race implied that Bill lost the race. People often become upset after failing to fulfill a goal. Since Bill failed to win, the reader can infer that he became upset for this reason.

Of course, a reader of stories (9) and (10) might be able to understand them by inferring the goals omitted from these stories. Then the reader could find the same explanations for the events in these stories events that were readily available in stories (1) and (3). Thus stories (10) and (9) could be understood by inferring that a character

had a goal in opposition to the goal already known in the story. In any case, these stories cannot be explained without understanding how the goals of one character could affect the goals of another.

### 8.1.1 Goal Competition

The relationship between the different characters' goals in stories (1) and (3) is called goal competition. Goal competition is a situation in which different characters have goals such that the fulfillment of one character's goal precludes the fulfillment of another's. For example, in story (1) John has the goal of enjoying the football game, and Mary has the goal of enjoying the ballet. These goals are in competition because each goal requires use of the same functional object, the television set, at the same time. In story (3), John's goal is to win the race, and Bill's goal is also to win the race. These goals compete because two different people cannot both be winners of the same race.

In story understanding, a reader must detect competing goals if he is to understand a character's behavior with respect to the competing goal. For example, suppose that after reading story (1), a reader were asked the question

Q1) What would have happened if Mary hadn't wanted to watch the ballet?

To respond with the answer

A1) John would have been able to watch the football game.

the reader has to infer that John failed to fulfill this goal because Mary was using the television set for a different purpose. To recognize that Mary's actions are at odds with John's intentions, the reader has to realize that Mary's goal was in competition with John's. If two characters' goals are in competition, and one of the characters fulfills a goal, an inference is that the other character will fail to fulfill his. Since Mary fulfilled her goal of watching the ballet, the reader can infer that John failed to fulfill his goal of watching the football game.

## 8.2 Detecting Goal Competition

To understand a story in which goal competition explains a character's behavior, a reader must first recognize that some character's goal competes with someone else's. Recognizing goal competition requires the reader to compare a goal of which it becomes aware against the known goals of the other story characters. Making this comparison presumes that the reader knows about the various ways goals can compete with one another, and can use this knowledge to determine whether the goals it is comparing are in competition.

The classification of goal competition situations needed for this task is almost identical to the the classification used in Chapter 6 to detect goal conflicts among the goals of the same character. Recall that this classification was as follows:

Types of Goals Competition

1. Resource Limitation

   The plans chosen for the goals share a common resource, and there is an insufficient quantity of that resource available for both plans.

2. Mutually Exclusive States

   The states two goals are intended to achieve are incompatible.

3. Generating a Preservation Goal

   The plan for one goal would cause the generation of a preservation goal.

Although the overall classification is the same for both goal conflict and goal competition, there are a number of differences in the way in which the *knowledge* is applied. These differences are now described for each type of goal competition.

8.2.1 Resource Limitations

A limitation of resources exists when there is a set of goals, all of which require a common resource, and that resource exists in a quantity that is sufficient for some but not all of the plans to succeed. For example, the following stories contain goal competition caused by a shortage of resources:

(11) John and Bill were driving across the desert when their car broke down. They had only enough water left for one of them to make it to the nearest town.

(12) John wanted to buy a stereo, but his wife Mary wanted to buy a television, and they had only enough money for one.

(13) John wanted to watch the football game on television, but Mary wanted to watch the Bolshoi ballet.

In story (11), both John and Bill require water in order to survive in the desert. Since the amount of water available is less than is needed for both of them to survive, John and Bill have preservation

goals that compete with one another due to a shortage of the resource water. The scarce resource in story (12) is money. Here both John and Mary have control of the same money, but want to use it for different purposes. In story (13) John and Mary both want to use a functional object for the same purpose at the same time. Since a television is not capable of projecting two programs simultaneously, John's and Mary's respective goals are in competition.

As was the case for goal conflicts, goal competition due to resource limitations is usually hinted at in the story text. However, hints themselves are somewhat different. The following events usually indicate that a limitation of resource is causing goal competition:

1. A statement that more than one planner wants to use the same functional object at the same time for different goals.

2. A statement that more than one planner wants to use a consumable resource that they share for different goals.

3. A statement that there is a lack of a consumable resource shared by several planners.

The primary difference between these criteria and those used for detecting goal conflicts based on resource shortages has to do with who can access the resource. In goal competition situations, a resource must be available to a number of planners at once. For example, it is possible to have goal conflict based a shortage of time, but not goal competition. Each character has his own time to use, and cannot take time away from another character. Thus an important part of detecting a goal competition situation is determining that all the planners have access to the same resource.

#### 8.2.1.1 Determining if a Resource is Shared

To determine if a resource can be accessed by several planners, I found it necessary to define the notion of a <u>shared resource</u>. A shared resource is a functional object that can be used by a number of planners. Recall that a functional object is one that is normally used in a plan. These are described in detail in Chapter 11 and were used extensively in the previous chapter.

There are two criteria that can qualify an object as a shared resource. These are as follows:

1. The object is owned by a group of planners. The group can be either an arbitrary assortment of people of a fixed social structure. For example, the money owned by a husband and wife is a shared resource because it jointly owned by both planners. A communally owned well is a shared resource since all the members of the community may draw water from it. Tennis courts owned by a private club are resource shared by the club members.

2. The object is not owned by anyone. Air is an example of a shared resource that has no owner.

To determine if a resource is shared by a group of planners, an understander can use the algorithm diagrammed in Figure 8.

Figure 8
Determining If a Resource is Shared by a Group of Planners

---

```
STEP 1:
 ______________________________         ____________________
|                              |       |                    |
| Is the resource marked as    |  yes  | The resource is    |
| never having an owner?       | ====> | shared.            |
|______________________________|       |____________________|

              ||
              ||  no
              \/
STEP 2:
 ______________________________         ____________________
|                              |       |                    |
| Is the resource marked as    |  no   | The resource is    |
| being owned by a group?      | ====> | not shared.        |
|______________________________|       |____________________|

              ||
              ||  yes
              \/
STEP 3:
 ______________________________         ____________________
|                              |       |                    |
| Are all the planners members |  no   | The resource is    |
| of that group?               | ====> | not shared.        |
|______________________________|       |____________________|

              ||
              || yes
              \/
       _________________
      |                 |
      | The resource is |
      | shared.         |
      |_________________|
```

---

Step 1 of this algorithm checks to see if the object is one that is never owned by someone. Such objects must be marked in memory as never having owners. For example, the representation for "air" in memory must contain the following information:

AIR

FUNCTION:

```
                                         |---> LUNGS
ONE <=> INGEST <--O-- AIR ---|
                                         |---<
```

CONSEQUENCES:

AIR IS STALE

OWNER:

NOBODY

That is, air is a functional object that is consumed as it is used because it becomes stale. The owner of air is marked as NOBODY, a symbol used to denote that the object cannot be owned. Thus the first step in the algorithm above merely checks to see if the owner slot of the object is filled with the symbol NOBODY.

Step 2 checks to see if the object is owned by a group. This requires examining the owner of the particular object in question. For example, a particular tennis court might be stored with the following information:

TENNIS-COURT-1

CLASS: OBJECT

TYPE: TENNIS-COURT

OWNER: CLUB-1

TENNIS-COURT-1 is a particular tennis court owned by CLUB-1. CLUB-1 is a group composed of some members. Step 3 of the algorithm would check to see if the planners being tested are members of this group. This can be done by checking to see if the planners are explicitly marked as being members of this group.

Another example of a group that shares resources is the family. For example, John, Mary and their child Junior might constitute a family called FAMILY-1. FAMILY-1 might then be used to represent the owner of objects held in common by that family. For example, we might have

TELEVISION-1

CLASS: OBJECT

TYPE: TELEVISION

OWNER: FAMILY-1

FAMILY-1

CLASS: GROUP

TYPE: FAMILY

MEMBERS: (JOHN, MARY, JUNIOR)

The algorithm in Figure 8 will find the TELEVISION-1 to be a shared resource for John, Mary and Junior since it is owned by FAMILY-1, and John, Mary and Junior are all members of that group.

### 8.2.2 Mutually Exclusive States

Characters' goals can also compete because achieving them would require mutually exclusive states to come into existence. For example, consider the following stories:

(14) John wanted to win the race. So did Bill.

(15) John wanted to marry Mary, but Mary wanted to marry Bill.

(16) John wanted to sell the sofa when they moved, but Mary wanted to take it with them.

(17) John wanted to wash the car, but his wife Mary wanted to take the car to the mechanic.

In story (14), the states that would constitute the fulfillment of John's and Bill's respective goals are exclusive. A race can only have one winner, so the achievement of either goal would preclude the achievement of the other. The same is true in story (15) where John's goal and Mary's goal compete because a person cannot be married to two people at the same time. In story (16), John's goal entails tranferring the sofa to another person. Since someone else owning the sofa would preclude Mary's goal of taking it with them, John's goal and Mary's goal are in competition. In story (17), John wants to wash his car, for which he needs control of the car. Mary also needs control of the car for her goal. Since the plans for both character's goals have exclusive preconditions, the goals are in competition.

The problem of determining if two states are mutually exclusive was discussed in the context of goal conflict in Chapter 6. Since the problem is identical in both cases, the solution presented in that chapter can be used here as well.

One particular form of goal competition based on mutually exclusive states merits special attention. Consider story (14) above:

(14) John wanted to win the race. So did Bill.

A race is something in which several people are in competition. That is, the idea of competition is part of the definition of a race. The goal of winning a race is an instance of an <u>inherently competitive goal</u>. An inherently competitive goal is a goal that cannot exist apart from competition with someone else's goal. Winning a race is inherently competitive because one cannot win a race without beating someone else who also wants to win. In contrast, story (15) has mutually exclusive goal states that are not inherently exclusive. Someone can want to marry someone else independently of whether there is another suitor, but one cannot have a race in which no one else is entered.

Processing inherently competitive goals does not require any special computation beyond that needed for other mutually exclusive goal types. I introduce the concept here only because it is a type of goal not introduced previously. An inherently competitive goal captures the intuitive notion of competition. It is the goal of beating someone else who has the same goal.

### 8.2.3 Generating a Preservation Goal

Consider the following stories:

(18) John asked Mary to stop smoking because the smoke annoyed him.

(19) John wanted to go to the track, but his wife didn't want him to go because she thought gambling was immoral.

(20) Johnny wanted an ice cream cone, but his mother thought it would spoil his appetite.

In each of these stories, one of the character's goals is caused by another character. Mary's smoking caused John to have the goal of eliminating the annoyance in story (18). John's desire to gamble induced his wife's goal of stopping him in story (19). Johnny's goal in story (20) caused his mother to want to insure Johnny didn't spoil his appetite. Each of these stories contained a goal competition situation in which one character's goal causes another character to have a preservation goal.

Detecting competition based on generating a preservation goal is identical to identifying goal conflict based on generating a preservation goal, except of course that the preservation goal is detected in another character.

## 8.3 Summary

There are a number of ways in which characters' goals can compete with one another. These can be classified using the same scheme developed for categorizing goal conflicts. The major difference between applying the scheme in each case is that to have goal competition based on a shortage of resources requires that the resources be shared by several planners.

The next chapter discusses the story situations to which goal competition can give rise.

# CHAPTER 9

## ANTI-PLANNING

## OR, UNDERSTANDING GOAL COMPETITION RESOLUTIONS

The previous chapter dealt with the problem of recognizing goal competition when it occurs in a story. This chapter is concerned with understanding a story once a goal competition situation is encountered. To explain the behavior of a character whose goals compete with those of another story character, it is useful to classify the situations arising from goal competition. A story reader needs to be capable of understanding several types of story situations. There are situations in which the characters pursue their goals while ignoring the competing goals of others. There are situations in which a character will try to undermine his opponent in order to achieve his own goal, or try to persuade his opponent to alter his goal or plan. There are situations in which a character can try to ease the goal competition. Lastly, there are situations in which the goal competition may be resolved by some external event.

The most interesting goal competition situations occur when a character explicitly tries to interfere with an opponent. These are called <u>anti-planning</u> situations. An anti-plan is a plan directed at stopping an opponent. There are three classes of planboxes that a characer may use in these situations: the OVERPOWER planbox, which means to interfere physically with one's opponent, the UNDO-PRECONDITION planbox, which applies to situations involving various forms of sabotage, and the Persuade package, in which one character tries to get another to change his mind.

Some particular anti-plans are useful for understanding goal competition based on preservation goals. For example, if a character has a plan he wishes to execute, but is afraid he will cause another character to have a goal whose pursuit will cause the first character to have a preservation goal, then the first character may use a plan to prevent this anticipated preservation goal. There are three planboxes applicable to this situation. In the AVOID-DETECTION planbox, the planner performs his original plan in such a way as not to be perceived

by the other character. The DISTRACT planbox may be used to direct someone's attention away from the planner's activity. Finally, the OVERPOWER planbox can be used to reduce the other character's ability to attend to the planner's actions, or to prevent him from being able to do anything about them.

## 9.1 Introduction

The previous chapter was concerned with the detection of goal competition. To this end, several classes of goal competition were distinguished. Each class required different kinds of knowledge and procedures by which a reader could identify the competition when it occurred in a story. It was shown that goal competition could occur in three ways: when there was a limited resource needed by the plans of several characters, when fulfilling the goals would lead to the existence of mutually exclusive states, and when the pursuit of one goal would cause another character to have a preservation goal.

Once a set of competing goals is detected in a story, a reader must understand how the competition affects each characters' goals and actions. For example, the following goal competition story was given in the previous chapter:

(1) John told Mary he wanted to watch the football game. Mary said that she wanted to watch the Bolshoi ballet.

John's goal competes with Mary's because both require use of the same functional object, the television set, at the same time and for different purposes. Consider the ways this story might be continued:

(2) Mary put on channel 3. John got out the lawnmower.

(3) John got the old black and white TV out of the attic.

(4) Mary put on channel 3. John punched Mary in the mouth and put on the ball game.

(5) John told Mary he would take her out to the ballet if she would watch the football game with him.

(6) Mary put on channel 3. She found out that the ballet was postponed until later that day.

Each of these continuations of story (1) represents a different goal competition situation. In every one of these situations, the reader can understand the events following the statement of the goal competition only in terms of that competition. That is, in each of the above stories, some character's behavior is directly related to the goal

competition. The reader must understand how goal competition can affect a person's behavior in order to find an explanation for that character's actions.

For example, in story (2), Mary pursues her goal while ignoring John's, and as a result, John ends up abandoning his goal altogether. Thus to understand why John got out the lawnmower, a reader needs to use the fact that John's goal competed with Mary's. That is, since Mary pursued her goal, which competed with John's, an action by John that was unrelated to his stated goal probably means that he gave up on this goal.

Instead of ignoring the competition in story (3), John takes some action to ease the competition. By getting the other television set, John and Mary should both be able to fulfill their respective goals. The reader needs to recognize the goal competition here in order to explain why John got a television set when one was already available.

John also takes the competition into account in his actions in story (4). However, in this story his behavior is somewhat more hostile. John eliminates Mary as a competitor by physically overpowering her, and then has his way with the television.

The fact that the characters' goals competes in story (4) is needed to explain why John struck Mary. John's goal was to watch the football game, and hitting someone is not usually part of a plan for this goal. John's action can be related to this goal not by a plan for fulfilling the particular goal, but through a plan that is applicable to goal competition situations. Thus the reader must have knowledge of these situations in order to understand how characters will act in them.

In story (5), John is somewhat more subtle. He tries to alleviate the competition by persuading Mary to abandon her goal in exchange for a favor. Again, the reader must understand this action in terms of the goal competition. Offering to take someone to the ballet is not usually part of a plan for watching a football game. Thus the reader must explain John's behavior in terms of how he was trying to deal with the competition.

Finally, in story (6), the goal competition dissolves when Mary learns that the time of the ballet has been changed. Since she no longer needs to use the television at the same time as John, their goals are no longer in opposition. The reader must be aware that the competition has vanished if the reader is to understand the possible continuations of the story.

For example, suppose (6) were continued with

(7) John watched the football game, and Mary got out the lawnmower.

The reader would not infer that Mary failed to watch the ballet, although this would be an appropriate inference if goal competition were present. But here the goal competition is resolved by the change in the schedule of the ballet. Since goal competition is no longer present,

Mary's action should be explained by ordinary planning rules, rather than by knowledge specific to goal competition. The reader probably infers the Mary was just pursuing another goal, rather than abandoning an old one.

As these examples indicate, there are a number of distinct alternatives that can be taken in a story with a goal competition. The possible courses that a story can follow fall into a number of categories, or goal competition situations. The classification of goal competition situations that I present below is based on the kinds of processing and knowledge that are required to understand stories. The purpose of this classification is to organize the knowledge and inference procedures needed to understand a story once a goal competition situation has been introduced. I found the following classes to be useful for explaining the behavior of characters who find their goals to compete with those of other story characters:

Goal Competition Situations

1. Independent goal pursuit. In this situation, each character tries to fulfill his own goal, ignoring the fact that his goal competes with someone else's. Some characters may succeed, causing other characters' goals to fail. Story (2) above was an instance of independent goal pursuit. Mary performed the actions necessary to achieve her own goal without regard to John's. As a result, Mary fulfilled her goal and John abandoned his.

2. Anti-planning. Anti-planning is a term used in Schank and Abelson (1977) to describe situations in which a character's actions reflect the presence of other characters' goals. For example, in story (4), John tries to fulfill his own goal by eliminating the competition. In story (5), John tries to persuade Mary to abandon her goal. In an anti-planning situation, a character forms a plan to prevent another character from fulfilling his goal, or to prevent someone from interfering with his own plan. For example, John's punching Mary in story (4) can be understood as an anti-plan to eliminate Mary from competing with John for the use of the television.

3. Easing the competition. Instead of pursuing their own goals, one or more of the characters with competing goals can try to ease the competition. For example, in story (3) John got another television so that both programs could be viewed at the same time. This action removed one of the conditions that caused the goals to compete, so that both characters could then pursue their respective goals independently.

4. External competition removal. The goal competition can be eliminated by some action not taken by one of the story characters. This was the case in story (6), where the time of Mary's program had been changed. Her program no longer was to be aired concurrently with John's. This removed a condition that caused the competition, enabling independent goal pursuit.

The remainder of this chapter discusses the processing of each of these goal competition situations in detail.

## 9.2 Independent Goal Pursuit

Independent goal pursuit is a term used to refer to situations in which story characters pursue their goals without regard to competing goals. One example of this situation was given above:

(8) John told Mary he wanted to watch the football game. Mary said that she wanted to watch the Bolshoi ballet. Mary put on channel 3. John got out the lawnmower.

In this story, Mary pursues her goal of watching the ballet without regard to John's competing goal of watching the football game. That is, Mary's plan is exactly the plan she would use had there been no competing goal. John cannot now pursue his plan without interfering with Mary's plan. John chose not to confront Mary in story (8). Instead, he abandoned his plan altogether. Story (8) can be understood with the aid of the following rule:

Rule S1:

If

1. a set of characters have competitive goals, and

2. *one of those characters pursues a plan for his goal,*

then

the other characters may abandon the plans for their goals.

Rule S1 is used in story (8) to explain why John took out the lawnmower. John and Mary had competing goals, and Mary was pursuing a plan for her goal. Rule S1 suggests that John may abandon his plan for watching the game. Since taking out the lawnmower is not a plan for watching a football game, the reader infers that John has given up on this goal entirely, and is now pursuing an unrelated goal.

While John abandoned both his plan and goal in story (8), this need not always be the case in independent goal pursuit situations. For example, consider the following story:

(9) John told Mary he wanted to watch the football game on television. Mary said that she wanted to watch the Bolshoi ballet. John rushed out to buy a ticket to the game.

As in story (8), Mary's plan forced John to abandon his, But instead of abandoning his goal altogether, John decided to try a new plan for the goal. Another plan for watching an event is to be there in person. John is pursuing this plan in story (9) by trying to get a ticket to the game.

Thus Rule S1 suggests plan abandonment. If a character abandons a plan for a goal, he may either abandon the goal altogether as in story (8), or switch to a new plan if one is available, as in story (9).

Independent goal pursuit can also characterize a situation in which several characters pursue their competing goals at once:

(10) John and Bill were stranded in the desert. They had only enough water for one of them to make it to the nearest town. They shared the water, and both died of thirst.

In story (10), both characters pursue their goals, and as a result, neither of their goals is fulfilled. This situation is possible when the goal competition is caused by a limited consumable resource. Then both goals can fail when each character consumes a quantity of the resource insufficient for his own goal, and which leaves an insufficient quantity of the resource for the other planners. That is, we have the following rule:

Rule S2:

If

1. a set of characters have competitive goals,

and

2. the goals compete due to the scarcity of a shared consumable resource,

and

3. each character pursues a plan for his goal,

then

the plans of all the characters may fail.

The scarce resource in story (10) was water. The characters need this consumable resource in order to remain alive. Both characters pursued their plans of drinking the water, so Rule S2 suggests that both plans will fail. If a plan for a goal fails, then the goal itself may fail. In this case, failure to preserve one's health results in one's health state being lowered. Thus when John and Bill die at the end of this story, the reader can infer that they died for lack of water. Without Rule S2, a story understander would probably have some difficulty determining why John and Bill died of thirst soon after drinking water.

A third type of independent goal pursuit occurs in the following story:

(11) John and Bill were both entered in the high hurdles. Both ran as hard as they could. John won.

Both characters in story (11) try to pursue their goals without interfering with the other character. Only one character could achieve his goal, since the characters' goals were mutually exclusive. Story (11) is an instance of a story in which characters with competing goals all pursue their goals independently. But unlike the previous story in which all the characters failed, here it was possible for one of the characters to achieve his goal. This situation is possible only when the goals compete due to mutually exclusive states. If a plan consumes a resource, or requires the use of a functional object, then the characters cannot pursue their goals without getting in each others' way. But they can independently pursue exclusive states in which the achievement of first state precludes the achievement of the others. This is in accordance with the following rule:

Rule S3:

If

1. a set of characters have competitive goals,
and

2. the goals compete because of exclusive states,
and

3. each character pursues a plan for his goal,

then
one of the characters may achieve his goal and the others fail.

In story (11), both John and Bill pursued their respective goals independently. Since only one person can win the race, one of them fulfilled his goal, and the others failed.

Another application of this rule is in the following story:

(12) John wanted to marry Mary. So did Bill. John proposed to Mary. Bill proposed also. Mary decided to marry Bill.

John's goal competes with Bill's because each of them being married to Mary constitutes a pair of mutually exclusive states. Each pursues his goal and one of them succeeds, causing the other to abandon his plan.

In sum, there are three kinds of independent goal pursuit situations. In the first situation, one character pursues his goal and the other characters abandon theirs. In the second situation, each character pursues a plan for his goal, and as a result, all the plans

fail. In the third situation all the planners again pursue their plans, but this time one of them succeeds and the others fail.

### 9.3 Anti-planning

An anti-planning situation is one in which a character's plans can be explained only in terms of the goals with which his goals compete. The following are instances of such situations:

(13) John wanted to watch the football game but Mary wanted to watch the ballet. Mary put on channel 3. John punched Mary in the mouth and put on the ball game.

(14) John wanted to watch the football game but Mary wanted to watch the ballet. John told Mary he would take her out to the ballet if she didn't watch the ballet on TV.

(15) John and Bill both wanted to win the high hurdles. John slipped Bill a mickey right before the race.

(16) John and Bill both wanted to marry Mary. John began spreading nasty rumors about Bill.

To explain John's behavior in each of these stories, a reader must understand that John is executing a plan aimed at reducing the competitiveness of his rivals. For example, in story (13) John physically overpowers Mary in order to watch the football game. This plan was applicable in story (13) because control of the television was a precondition for both character's plans. Since overpowering someone is a way to remove that person's control over an object, John could overpower Mary to remove her control over the television. Having sole control over a resource whose shortage caused the goal competition assures the use of the object for that planner, and forces the other characters' plans to fail.

Story (13) embodies a technique applicable to all goal competition situations: eliminating the opposition. A plan that is directed at eliminating an opponent is called an <u>anti-plan</u>. These fall into three classes:

1. An opponent can be physically prevented from performing a plan. John punching Mary in story (13) was an instance of this technique. Punching Mary presumably reduced her ability to exercise her control over the television, thus enabling John to use it for his own purposes.

2. An opponent can be sabotaged. Sabotage consists of undoing one of the preconditions for another character's plan. For example, story (16) is a sabotage situation. John tries to sabotage Bill's chances of marrying Mary by getting Mary to think that Bill is an undesirable person. Being well thought of by one's intended spouse is a precondition for the usual plan for marrying someone, i. e.,

asking that person to marry you. Thus John is sabotaging Bill by undoing a precondition for his plan to wed Mary.

3. An opponent can be persuaded to abandon his plan. One's competitor can be persuaded to give up, or enticed into performing a plan for some other goal. Story (14) is an example of persuading someone to abandon a plan. John offers Mary something in exchange for her abandoning her current goal altogether. Then Mary would no longer compete with John for use of the television, and John could pursue his goal unhampered.

These methods of reducing the effectiveness of one's opponent are planning techniques specific to goal competition. That is, the last section showed what could happen if several planners pursued their goals while ignoring the fact that their goals competed. Here the planners are behaving in full recognition of this fact. In order to understand a story in which a character pursues his own goal by manipulating his opponent, the reader must identify the competition between the characters' goals, and then determine which of these anti-plans the planner is using. We now examine how a reader can understand each form of goal competition elimination.

### 9.3.1 Physical Elimination

Physical elimination is a plan applicable to all goal competition situations. Essentially, a planner can physically hamper his opponent by using the OVERPOWER plan in one of two ways. The plan can be used to reduce the physical abilities of the opponent, or interrupt the opponent's pursuit of his goals. For example, consider stories (13) and (15) above:

(13) John wanted to watch the football game. Mary wanted to watch the ballet. Mary put on channel 3. John punched Mary in the mouth and put on the ball game.

(15) John and Bill both wanted to win the high hurdles. John slipped Bill a mickey right before the race.

In each of these stories, John uses the OVERPOWER plan to physically reduce his opponents abilities. Punching Mary in story (13) presumably disables her for a while, and drugging Bill in story (15) will lower Bill's consciousness and thus adversely affect his ability to race.

To understand these situations, a story understander can use the following rule:

Rule S4:

If

1. a set of characters have competitive goals,
   and

2. one of the characters tries to physically disable another character,

then

that action is part of a plan to fulfill its actor's goal by eliminating the opposition.

For example, John and Bill have competitive goals in story (15), and John tries to physically disable Bill by drugging him. Thus Rule S4 states that John's action is part of a plan for his goal of winning the race.

Rule S4 implies that the reader can interpret a plan to counter someone else's goal as a plan for one's own goal. Of course, this is not strictly true. If a plan for a character's goal succeeds, then that character's goal is fulfilled. This is not true of plans for eliminating the opposition. John could successfully drug Bill in story (15), but still fail to win the race. The only plan that can actually fulfill John's goal is for him to run the race.

Thus there is a representational issue of how the use of a plan to eliminate opposition should be recorded in the story representation. It cannot be stored simply as a plan for its performer's own goal, since this might lead to the inference that if the plan succeeded that goal was fulfiled. Furthermore, eliminating the opposition is not a precondition for a plan for this goal, either. That is, John's plan in story (15) is to run the race, and this plan does not have a precondition of eliminating the opposition; the race can be run and won without so doing.

Hence we need a way to represent plans whose goal is to stifle the opposition. Recall that previously, the representation contained a link between a goal and a plan used for that goal. I now add to the representation a link between a goal and an anti-plan. Schank and Abelson used the notion of an anti-plan to describe the behavior of a character who has a preservation goal. Here I broaden the definition of anti-plan to include any plan directed at subverting an opponents goal. Figure 9 shows how story (15) can be represented with the use of anti-plans.

AD-A062 629 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 5/7

UNDERSTANDING GOAL-BASED STORIES.(U)

SEP 78 R WILENSKY N00014-75-C-1111

UNCLASSIFIED RR-140 NL

3 OF 4

ADA

062629

Figure 9
John tries to undermine Bill's plan.

```
           Goal - John win race              Goal - Bill win race
                  Outcome -                         Outcome -
                     /    \
                    /      \
                   /        Plan   - run race
                  /
        Anti-plan
          Goal - Bill IS PHYS-STATE(-)
            Outcome -
               |
               |
            Plan - OVERPOWER ---> action - John drug Bill
```

John's goal in Figure 9 has two links emanating from it. One link points to the plan for that goal, and the other to an anti-plan. The anti-plan structure is a typical goal episode. That is, it contains a goal and plans used to fulfill that goal. The goal in Figure 9 is for Bill to be in a negative physical condition. John chose to achieve this goal by using the OVERPOWER plan. This plan in turn gave rise to the action of John drugging Bill. The outcome of this goal, as well as that of John's and Bill's original goals, is not yet known at this point in the story.

Figure 9 represents a story in which the particular anti-plan used involved physical disablement. However this is a general format for representing any plan used to eliminate an opponent. Thus all the stories in this section conform to a structure similar to that shown in this figure.

A reader needs to be able to recognize one other situation in which force is used to hinder an opponent. Consider the following story:

> (17) John and Bill were neck and neck in the final stretch. John stuck his foot out and tripped Bill.

Tripping someone will not reduce that person's physically capabilities, as giving someone a drug will. Tripping is a way of physically interfering with someone's motion so as to interrupt the execution of that person's plan. Unlike story (15), John is trying to win the race not by reducing Bill's health state, but by using the OVERPOWER planbox to interrupt Bill's pursuit of his goal. This situation is similar to the previous one except that here the goal within John's anti-plan is the temporary reduction of Bill's control over his own body. Interfering with someone's bodily control can be used to hinder that person from successfully executing a plan requiring either bodily movement or use of one's body to control an object. That is, we have the following rule:

Rule S5:

If

1. a set of characters have competitive goals,
and

2. one character's plan requires him to move or to use his body to control an object,
and

3. another character tries to physically interfere with that character's control over his own body,

then
that action is part of a plan to fulfill its actor's goal by eliminating the opposition.

For example, Bill's plan in story (17) is to run the race, and that involves moving his body. Tripping someone is normally a plan for reducing that person's bodily control. Since John's goal competes with Bill's, Rule S5 suggests that John tripped Bill to eliminate him as competition.

9.3.2 Sabotage

I use the term sabotage to refer to those situations in which one character intentionally violates a precondition for another character's plan. For example, recall story (16):

(16) John and Bill both wanted to marry Mary. John began spreading nasty rumors about Bill.

John's behavior can be explained in terms of the competition between his goal and Bill's. If Bill fulfilled his goal of being married to Mary, then John would be unable to fulfill his goal. John is therefore using an anti-plan to stymie Bill's plan. I call this anti-plan UNDO-PRECONDITION. UNDO-PRECONDITION is a planbox in which a character ruins a precondition for another character's plan. For example, the usual plan for getting someone to marry you is to ask that person. For this plan to succeed, the person asked must be favorably disposed toward the person popping the question. That is, in story (16), Bill's plan for getting Mary to marry him has the precondition that Mary likes Bill. John has chosen to counter this plan by getting Mary to dislike Bill. People ordinarily don't like people who have done bad things. Thus John is trying to get Mary to not like Bill by getting her to believe that Bill has done such things.

John's behavior in story (16) can be accounted for with the following rule:

Rule S6:

If

1. a set of characters have competitive goals,

   and

2. one character does an action whose goal is to establish a state that excludes a precondition for the plan of another character,

then

that action is part of a plan to fulfill its actor's goal by eliminating the opposition.

For example, in story (16), John spreads nasty rumors about Bill. The goal behind spreading nasty rumors about a person is usually to get other people to dislike that person. Thus Rule S6 states that John's action is part of a plan to eliminate Bill as a competitor insofar as John's goal of marrying Mary is concerned.

Another application of Rule S6 is in the following story:

(18) John and Bill both wanted to win the stockcar race. Just before the race began, John cut Bill's ignition wire.

John's goal competes with Bill's since both of them want to win the race. In order to be in an auto race, one must have a car in good physical condition. By reducing the physical condition of Bill's car, that car no longer meets the requirements of Bill's plan. Thus rule S6 states that John must have cut Bill's ignition wire in order to eliminate him as a competitor.

As these examples indicate, sabotage is a very general anti-plan since all plans have preconditions that can be violated. However, there are a number of standard forms that sabotage often takes. These include impairing a functional object required by another planner, using up a consumable required by another planner, and getting other people to have negative attitudes toward another planner. For example, story (18) is a case of impairment of a functional object, and story (16) is an instance of generating negative attitudes.

The knowledge that these plans are typically used for sabotage is useful for inferring a goal competition situation that may not be otherwise detected. For example, consider this story:

(19) John learned that Bill wanted to marry Mary. John began spreading nasty rumors about Bill.

In story (19) the reader is not told what John's goal is. However, when the reader learns that John is spreading rumors about Bill, the reader can use the knowledge that this is often a type of sabotage to see if John could be trying to prevent Bill from achieving a goal. Since John

has just learned that Bill wants to marry Mary, the reader can check to see if John's action sabotages this goal. As above, the goal behind this action is found to violate a precondition for Bill's plan to marry Mary. Thus John appears to be sabotaging Bill. One reason for sabotaging someone is to prevent them from achieving a goal that competes with your goal. Thus the reader could infer that John has a goal that competes with Bill's.

9.3.2.1 Computer Example -

The following is a trace of PAM being run on story (18) above.

[PHOTO: Recording initiated Thu 17-Aug-78 2:13AM]

@RUN PAM

*(UNDERSTAND CD12)

THE STORY IS

JOHN WANTED TO WIN THE STOCKCAR RACE.
BILL ALSO WANTED TO WIN THE STOCKCAR RACE.
BEFORE THE RACE JOHN CUT BILL'S IGNITION WIRE.

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(JOHN WANTED TO WIN THE STOCKCAR RACE) | |
| CONCEPTUALIZATION IS:<br>((CON<br>((ACTOR HUMO IS<br>(*WINNER*<br>COMPETITION ACTIVO))<br>TIME (FORM11))<br>IS (*GOAL* PART HUMO))<br>TIME (FORM12)) | ACTIVO denotes the stockcar race that HUMO (John) wants to win. |
| NOT A PREDICTED INPUT | |
| BEGIN SEARCH FOR EXPLANATION | |

| | |
|---|---|
| TESTING EXPLANATION OFFERED BY AWINNING-REQ<br><br>EXPLANATION IS GOAL:<br>(*AWIN* PLANNER HUMO COMPETITION ACTIVO)<br>PLAN: ($AUTORACE PLANNER HUMO RACEOBJ PHYS1)<br><br>WITH SOURCE (*LIFE-THEME* TYPE (*COMPETITIVE*))<br><br>EXPLANATION CONFIRMS PREDICTION INIT-REQ<br><br>FOUND EXPLANATION SEQUENCE:<br><br>AWINNING-REQ | This request encodes the information that winning a race is an inherently competitive achievement goal. That is, the goal is to compete against other people with exclusive goals. This request also asserts that the plan for this goal is to do the racing script using a car, and that John has this goal simply because he is competitive. |
| *** ADDING TO STORY REPRESENTATION:<br><br>LOADING PREDICTION WIN-GOAL-COMP-REQ<br><br>INFERRED GOAL: (*AWIN* PLANNER HUMO COMPETITION ACTIVO) | Because the goal is inherently competitive, PAM loads a request that will find this goal to be in competition with anyone else who also has the goal of winning this event. |
| NEXT INPUT IS:<br>(BILL ALSO WANTED TO WIN THE STOCKCAR RACE)<br><br>CONCEPTUALIZATION IS:<br>((CON<br>((ACTOR HUM1 IS<br>(*WINNER* COMPETITION ACTIVO))<br>TIME (FORM21))<br>IS (*GOAL* PART HUM1))<br>TIME (FORM22))<br><br>NOT A PREDICTED INPUT<br><br>BEGIN SEARCH FOR EXPLANATION<br><br>TESTING EXPLANATION OFFERED BY AWINNING-REQ | |

| | |
|---|---|
| EXPLANATION IS GOAL:<br>(*AWIN* PLANNER HUM1 COMPETITION<br>ACTIVO)<br>PLAN: ($AUTORACE PLANNER HUM1<br>RACEOBJ PHYS2)<br><br>WITH SOURCE (*LIFETHEME* TYPE<br>(*COMPETITIVE*))<br><br>EXPLANATION CONFIRMS PREDICTION<br>INIT-REQ<br><br>FOUND EXPLANATION SEQUENCE:<br><br>AWINNING-REQ<br><br>*** ADDING TO STORY REPRESENTATION:<br><br>LOADING PREDICTION<br>WIN-GOAL-COMP-REQ<br><br>INFERRED GOAL: (*AWIN* PLANNER HUM1<br>COMPETITION ACTIVO)<br><br>***** GOAL COMPETITION DETECTED | The initial processing for this sentence is the same as for the first. |
| COMPETITION IS BETWEEN<br>GOAL: (*AWIN* PLANNER HUMO<br>COMPETITION ACTIVO)<br>AND<br>GOAL: (*AWIN* PLANNER HUM1<br>COMPETITION ACTIVO)<br><br>***** COMPETITION IS BASED ON<br>MUTUALLY EXCLUSIVE GOAL STATES | The request loaded above looking for a competing goal goes off, since both John and Bill have the goal of winning the same race. Goal competition based on mutual exclusion is inferred. |
| INPUT CONFIRMS PREDICTION<br>WIN-GOAL-COMP-REQ<br><br>LOADING PREDICTION<br>UNDO-PRECOND-REQ<br><br>LOADING PREDICTION<br>ELIM-OPPOSITION-REQ<br><br>LOADING PREDICTION<br>PERSUADE-OPPOSITION-REQ | Now several requests are loaded, looking for various situations that might follow this form of goal competition. UNDO-PRECOND-REQ looks for instances of one character sabotaging the other, ELIM-OPPOSITION-REQ looks for one character to overpower the other, and PERSUADE-OPPOSITION-REQ looks for one character to convince the other not to compete. |

```
NEXT INPUT IS:
 (BEFORE THE RACE JOHN CUT BILL'S
  IGNITION WIRE)

CONCEPTUALIZATION IS:
((CON
  ((ACTOR HUMO <=> (NIL))
   TIME (FORM33))
  LEADTO
  ((ACTOR PHYS3 TOWARD
    (*PSTATE* VAL (-10.)))
   TIME (FORM34))))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
ADESTROY-REQ

EXPLANATION IS GOAL:
(*ANEGSTATE* PLANNER HUMO OBJECT
  PHYS3)
 PLAN: (*UNSPEC* PLANNER (NIL))
```

Since cutting a functional object whose use does not require cutting usually means that the object is being damaged, PAM infers from the input that John has the goal of damaging the ignition wire. Now an explanation for this goal is sought.

```
EXPLANATION CONFIRMS
PREDICTION UNDO-PRECOND-REQ

FOUND EXPLANATION SEQUENCE:

ADESTROY-REQ
```

The prediction UNDO-PRECOND-REQ is able to explain this goal. Damaging a part of an object is a way to damage an object, and an ignition wire is part of a car. Since a car is needed to perform the racing script, John's action is found to be a way of undoing a precondition for Bill's plan.

```
*** ADDING TO STORY REPRESENTATION:

INPUT CONFIRMS PREDICTION
UNDO-PRECOND-REQ

INFERRED GOAL:
(*ANEGSTATE* PLANNER HUMO OBJECT
 PHYS3)
WITH ANTIPLAN
 PLAN: (*PB-UNDO-PRECOND* PLANNER
HUMO NEUTRALIZEDPLAN
($AUTORACE PLANNER HUM1 RACEOBJ
 PHYS2)

FINISHED UNDERSTANDING PHASE
```

Thus PAM infers that John wanted to damage Bill's car as part of an anti-plan. The anti-plan is to undo a precondition for Bill's plan of doing the racing script using his car.

*(DOQA)

QUESTION: ?Q1

Why did John break an ignition wire?
Because he was trying to prevent Bill from racing.

[PHOTO: Recording terminated Thu 17-Aug-78 2:15AM]

To answering question Q1, the explanation for John's cutting the ignition wire is found. The explanation states that this action is part of an anti-plan to undo a precondition for Bill's plan. This answer is an inference PAM made based on noticing the goal competition between the two characters.

### 9.3.3 Avoiding danger

A special class of anti-planning rules is needed to understand goal competition situations based on generating a preservation goal. As was discussed in the previous chapter, one form of goal competition occurs when the pursuit of one character's plan generates a preservation goal for another character. There are two classes of situations in which this kind of goal competition arises: situations in which a planner is pursuing a preservation goal, and situations in which a planner is trying to prevent someone from having a goal whose pursuit would cause him to have a preservation goal.

For example, consider the following stories:

(20) John saw his enemy Bill down below. John pushed a boulder down the mountain towards Bill. Bill saw the boulder coming and jumped out of the way.

(21) John wanted to get the treasure, but first he had to get past the dragon. John tiptoed past the dragon and picked up the treasure.

Story (20) contains an instance of planning for a preservation goal. That is, John's plan threatens Bill with physical harm, and thus Bill probably has the preservation goal of preserving his well-being. Since John's plan entailed that Bill be in the way of the boulder when it reached Bill, Bill was able to foil John's plan and undo this precondition by moving out of the way.

Story (21) is an instance of a story in which a character is trying to execute a plan without causing another character to have a goal that would then compete with his own. John wants to execute a plan to get the treasure, but he must do so in such a way so as not to cause the dragon to want to kill him. John therefore tries to execute his plan without drawing the attention of the dragon.

9.3.3.1 Fulfilling preservation goals -

Fulfilling a preservation goal is not strictly an anti-planning task. Preservation goals may be induced by natural dangers as well as man-made ones. The rules that follow are applicable to both types of situations when the threat is of a physical nature. For example, some of the ways to avoid physical danger are the following:

Rule PHYS1: If the danger involves being contacted by a moving object, then move out of the path of that object.

Rule PHYS2: If the danger involves being contacted by a moving object, then interpose an object between yourself and the threatening object.

Rule PHYS3: If the danger involves being contacted by a moving object, then stop that object or change its course.

Rule PHYS4: If the danger involves being near a dangerous object, then move away from that object.

For example, Rule PHYS2 could be used to understand story (22):

(22) John was walking along when it began to rain. John got under an awning.

The "danger" here is becoming wet from the rain. The rain is a moving object, so Rule PHYS2 states that the danger can be prevented by interposing an object between the rain and the planner. Since moving under an awning constitutes such an action, a reader could explain John's action as an instance of a plan for his goal of preserving his comfort.

Many of these rules have related, script-like instantiations. For example, opening an umbrella is a script-like method for interposing an object between oneself and the rain. A reader needs to know a number of such script-like plans associated with very particular preservation goals. For example, getting out of a house during an earthquake is another script-like plan for a naturally induced preservation goal.

The plans for avoiding danger from natural causes are also applicable to some dangers that are caused by people. A person can harm another person by directing a force at that person. For example, it is possible to injure someone by throwing something at that person and to make someone uncomfortable by throwing water at him. Since these plans involve changing someone's physical state, they can be planned against using the above rules. For example, if John tried to throw a knife at Bill, Bill could avoid the danger by stepping out of the way in accordance with Rule PHYS4. Likewise, a reader could understand why John stepped under an awning if Bill threw some water at him by applying Rule PHSY2.

Thus the above rules are rules for countering physically undesirable situations. When a natural force is involved, then these are the only rules that are applicable. If an animate actor is involved, then these rules can be used along with rules for anti-planning against another character's intentions.

9.3.3.2 Anticipated preservation goals -

A planner may wish to execute a plan so as to avoid causing someone else to have a goal in competition with his own. This situation may occur when the execution of a plan will put the planner in the way of a natural danger, or when it will cause another planner to have a negative intention toward him. For example, recall story (21) above:

(21) John wanted to get the treasure, but it was guarded by a huge dragon. John tiptoed past the beast.

In story (21), John apparently realizes that if the dragon becomes aware of his presence, he will try to hurt John. Thus John must execute his plan to get by the dragon without the dragon becoming aware of his presence.

Story (21) contains an instance of what I call an _anticipated preservation goal_. That is, the planner is not responding to an actual threat, but trying to prevent a situation from arising that would necessitate his doing so. Thus the anti-plans that are applicable to anticipated preservation goals are plans for preventing another character from learning of one's intentions.

There are three planboxes applicable to anticipated preservation goals:

1. AVOID-DETECTION

2. DISTRACT

3. OVERPOWER

AVOID-DETECTION is a plan for doing any action in such a way so as to prevent another planner from attending to your actions. Since the other planner will be able to thwart a plan only upon detecting its execution, this planbox can be used to execute a plan for one goal without invoking an undesirable goal in someone else. This conforms to the following rule:

Rule ANTICIPATE1:

If

1. A character wishes to perform an action

and

2. that action might cause another character to have a goal

and

3. that goal will cause the first character to have a preservation goal,

and

4. the first character performs his desired action in such a way so as to prevent the other character from detecting it,

then

performing the action in that manner is an instance of using the AVOID-DETECTION plan for the anticipated preservation goal to which the action could have given rise.

For example, in story (21), tiptoeing past the dragon is a way of executing the SELF-TRANSPORT plan for the goal of being near the treasure, while at the same time preventing the dragon from learning of the action, and thus having the goal of hurting John. According to Rule ANTICIPATE1, performing the SELF-TRANSPORT plan in a quiet manner is an instance of using the AVOID-DETECTION planbox for the anticipated preservation goal of staying healthy. Thus an application of Rule ANTICIPATE1 in story (21) would cause the story reader to make the inference that John didn't want to attract the attention of the dragon in order to ensure that the dragon wouldn't try to hurt him.

The AVOID-DETECTION planbox can also be used when the planner wishes to avoid disturbing a natural force. For example, consider the following story:

(23) John wanted to ski down the slope, but he was afraid he would cause an avalache. John crept down the slope very cautiously.

This story is similar to story (21), except that here the danger is not animate. That is, John uses the AVOID-DETECTION planbox in story (23) to execute his other plan in such a way so as to reduce the physical side-effects of his actions as much as possible. This type of story conforms to a rule similar to Rule ANTICIPATE1:

Rule ANTICIPATE2:

If

1. A character wishes to perform an action

and

2. that action might cause a natural force to act

and

3. that action taken by the natural force will cause the character to have a preservation goal,

and

4. the first character performs his desired action in such a way so as to minimize the physical side-effects of action,

then

performing the action in that manner is an instance of using the AVOID-DETECTION plan for the anticipated preservation goal to which the action could have given rise.

The DISTRACT plan calls for the planner to cause his opponent to have his attention elsewhere while he executes his plan. This prevents the opponent from attending to the planner and therefore having a goal the planner might find disagreeable. Thus DISTRACT applies to anticipated preservation goals according to this rule:

Rule ANTICIPATE3:

If

1. A character wishes to perform an action

and

2. that action might cause another character to have a goal

and

3. that goal will cause the first character to have a preservation goal,

and

4. the first character tries to get the second character to attend to something other than the first character's actions,

then

the first character is using the DISTRACT plan for the anticipated preservation goal to which the action could have given rise.

For example, John could throw a rock on one side of the dragon, and then try to slip past the dragon on the other side as the dragon goes to investigate the cause of the disturbance. An understander with Rule ANTICIPATE3 could explain John's behavior as a plan to distract the dragon while he executes his plan.

The third way to fulfill an anticipated preservation goal is by using the OVERPOWER planbox to reduce an opponent's ability to attend to one's activity. For example, John could slip a sleeping potion into the dragon's water in order to OVERPOWER the dragon. Then John could walk past the dragon without fear that the dragon would have the goal of hurting him. This is in accordance with the following rule:

Rule ANTICIPATE4:

If

1. A character wishes to perform an action

and

2. that action might cause another character to have a goal

and

3. that goal will cause the first character to have a preservation goal,

and

4. the first character takes some action to reduce the physical abilities of the second character,

then

the first character is using the OVERPOWER planbox for the anticipated preservation goal to which the action could have given rise.

9.3.3.2.1 A Computer Example -

This section contains a trace of PAM processing a version of story (21).

```
[PHOTO:  Recording initiated  Fri 25-Aug-78 10:39AM]

@RUN PAM

(UNDERSTAND CD13)

THE STORY IS

 JOHN WANTED TO GET THE TREASURE,
 BUT IT WAS GUARDED BY A DRAGON.
 JOHN WALKED OVER TO THE TREASURE QUIETLY.
```

| COMPUTER OUTPUT | ANNOTATION |
|---|---|

```
PROCESSING ...

NEXT INPUT IS:
 (JOHN WANTED TO GET THE TREASURE)

CONCEPTUALIZATION IS:
((CON
  ((ACTOR MONEY0 IS
    (*POSS* PART HUM0))
   TIME (FORM11))
  IS (*GOAL* PART HUM0))
 TIME (FORM12))
```

John has the goal of controlling a valuable object (MONEY0). PAM processes this input just as it does the change of control goals that appeared in previous stories. See the output in Chapter 2 for a description of the processing.

```
NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
DCONT-EPISODE-REQ

EXPLANATION IS GOAL:
(*DCONT* PLANNER HUM0 OBJECT MONEY0
OWNER HUM1 RECIPIENT HUM0)

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
USE-MONEY-REQ

EXPLANATION IS GOAL:
(*DCONT* PLANNER HUM0 OBJECT MONEY0
 OWNER HUM1 RECIPIENT HUM0)

EXPLANATION CONFIRMS PREDICTION
INIT-REQ

FOUND EXPLANATION SEQUENCE:

DCONT-EPISODE-REQ -> USE-MONEY-REQ
```

```
*** ADDING TO STORY REPRESENTATION:

LOADING PREDICTION
DCONT-ATTEMPT-REQ

 GOAL: (*DCONT* PLANNER HUMO OBJECT
MONEYO OWNER HUM1 RECIPIENT HUMO)

LOADING PREDICTION
FOND-THEME-REQ

LOADING PREDICTION
DO-BUY-PLAN-REQ

LOADING PREDICTION
SUBSUM-REQ

NEXT INPUT IS:
 (BUT IT WAS GUARDED BY A HUGE
  DRAGON)

CONCEPTUALIZATION IS:
((CON
  ((ACTOR MONEYO IS
    (*POSS* PART HUM1))
   TIME (FORM17))
  IS (*GOAL* PART HUM1))
 TIME (FORM20))
```

This represention shows that the dragon wants to control the treasure. In addition, the representation for the treasure includes the fact that the treasure is already controlled by the dragon. Thus PAM interprets this input as a statement of the dragon's goal of preserving his control over the treasure.

```
NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
PCONT-EPISODE-REQ


EXPLANATION IS GOAL:
(*PCONT* PLANNER HUM1 OBJECT MONEYO
OWNER HUM1 RECIPIENT HUM1)

 WITH SOURCE (*PRESERVATION*)

EXPLANATION CONFIRMS PREDICTION
INIT-REQ

FOUND EXPLANATION SEQUENCE:

PCONT-EPISODE-REQ
```

```
*** ADDING TO STORY REPRESENTATION:

 GOAL: (*PCONT* PLANNER HUM1 OBJECT
MONEYO OWNER HUM1 RECIPIENT HUM1)

***** GOAL COMPETITION DETECTED

COMPETITION IS BETWEEN
 GOAL: (*DCONT* PLANNER HUMO OBJECT
 MONEYO OWNER HUM1 RECIPIENT HUMO)
AND
 GOAL: (*PCONT* PLANNER HUM1 OBJECT
MONEYO OWNER HUM1 RECIPIENT HUM1)

***** COMPETITION IS BASED ON
MUTUALLY EXCLUSIVE GOAL STATES
```

When this goal is added to the story representation, PAM tries to determine its relationship to the goal it inferred from the previous sentence. Since a change of control of an object excludes the possibility that someone else can preserve control over that object, PAM detects goal competition based on mutually exclusive goal states.

```
LOADING PREDICTION
UNDO-PRECOND-REQ

LOADING PREDICTION
ELIM-OPPOSITION-REQ

LOADING PREDICTION
PERSUADE-OPPOSITION-REQ

LOADING PREDICTION
AVOID-DETECTION-REQ
```

PAM loads a number of predictions pertaining to different forms of anti-planning.

```
NEXT INPUT IS:
 (JOHN WALKED OVER TO THE TREASURE
  QUIETLY)

CONCEPTUALIZATION IS:
((ACTOR HUMO <=> (*PTRANS*)
  OBJECT HUMO
  TO (*PROX* PART MONEYO)
  INST ((ACTOR HUMO <=> (*MOVE*)
         OBJECT PARTO)
        TIME (FORM31)))
 MANNER ((*QUIETLY*)) TIME (FORM33))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION
```

PAM looks for an explanation for this input, and finds two:

TESTING EXPLANATION OFFERED BY
WALK-GOAL-EPISODE-REQ

EXPLANATION IS GOAL:
(*DPROX* PLANNER HUMO OBJECT HUMO
LOCATION (*PROX* PART MONEYO))
PLAN: (*PB-WALK* PLANNER HUMO
LOCATION (*PROX* PART MONEYO))

First, in "standard" bottom-up fashion, PAM interprets the input as part of a plan to be near the treasure for the purpose of taking it.

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
DO-TAKE-REQ

EXPLANATION IS
PLAN: (*PB-TAKE* PLANNER HUMO
OBJECT HUMO)

EXPLANATION CONFIRMS PREDICTION
DCONT-ATTEMPT-REQ

FOUND EXPLANATION SEQUENCE:

WALK-GOAL-EPISODEREQ -> DO-TAKE-REQ

*** ADDING TO STORY REPRESENTATION:

INPUT CONFIRMS PREDICTION
DCONT-ATTEMPT-REQ

PLAN: (*PB-WALK* PLANNER HUMO
LOCATION (*PROX* PART MONEYO))

This explanation is added to the story representation.

INFERRED GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*PROX*
PART MONEYO))

OUTCOME OF GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*PROX*
PART MONEYO)) IS (*SUCCEED*)

INPUT CONFIRMS PREDICTION
AVOID-DETECTION-REQ

An additional explanation is found by one of the predictions made above based on goal competition. AVOID-DETECTION-REQ predicted that John might try to avoid generating a preservation goal in pursuit of his change of control

| | |
|---|---|
| INFERRED GOAL:<br>(*PKNOW* PLANNER HUM0<br>FACT<br>((ACTOR HUM0 IS (*PROX* PART<br>(*UNSPEC* CLASS (*LOCATION*)))))<br>PREVENTEE HUM1)<br>WITH ANTIPLAN<br>PLAN: (*PB-AVOID-DETECTION* PLANNER<br>HUM0 PREVENTEE HUM1) | goal. Thus PAM infers that John wants to prevent the dragon from knowing John's location, and that he is using the AVOID-DETECTION planbox for this task. |
| LOADING PREDICTION<br>SUITABLE-SUBGOAL-REQ<br><br>LOADING PREDICTION<br>RESULT-IN-DCONT-REQ | PAM goes on to load predictions that would be useful for understanding subsequent events if the story were to continue. |
| FINISHED UNDERSTANDING PHASE | |

---

```
*(DOQA)

QUESTION: ?Q1

 Why did John sneak over to the treasure?
 Because he was trying to get the treasure and he wanted to
prevent the dragon from knowing where he was.

[PHOTO:  Recording terminated  Fri 25-Aug-78 10:41AM]
```

To answer question Q1, PAM consults the story representation to find the reasons for this event. Two reasons are found, the first stating that this is part of a plan to take possession of the treasure, and the second describing the event as part of a plan to prevent his detection by his presumed opponent. These reasons are expressed as the answer to the question.

### 9.3.4 Persuading an Opponent

The third way in which a planner can eliminate an opponent is by persuading that opponent to abandon his plan, or to pursue some other goal. For example, recall story (14) above:

> (14) John wanted to watch the football game but Mary wanted to watch the ballet. John told Mary he would take her out to the ballet if she didn't watch the ballet on TV.

This story contains the familiar goal competition between John and Mary based on their need to use the same functional object at the same time. Here John tries to bargain with Mary to get her to abandon her

goal of watching the ballet, and thus eliminating her comp tition for the television. In principle, John could have used any of the plans in the persuade package to try to convince Mary not to watch the ballet on television. Thus we have the following rule:

Rule S7:

If

1. a set of characters have competitive goals,
and

2. one character tries to persuade another character to abandon his plan or goal,

then

the persuasion is part of a plan to fulfill its actor's goal by eliminating the opposition.

In story (14), John tried to persuade Mary to abandon her goal, so Rule S7 suggests that this is an anti-plan to enable John to watch the football game.

In addition to these straightforward requests, a reader needs to be able to interpret the suggestion that a character pursue an alternate plan or goal. For example, consider this story:

(24) John wanted to watch the football game but Mary wanted to watch the ballet. John asked Mary if she would watch the ballet at her mother's.

This story is almost identical to story (14) except that here John is trying to persuade Mary to adopt a different plan. To recognize that this is a way of persuading Mary to abandon her competing plan, a story reader would need the following rule:

Rule S8:

If

1. a set of characters have competitive goals,
and

2. one character tries to persuade another character to perform a plan that does not compete with his plan,

then

the persuasion is part of a plan to fulfill its actor's goal by eliminating the opposition.

In story (24), John asks Mary to perform a plan that does not require use of the same functional object as John plans to use. Since this plan would not compete with John's plan, Rule S8 would find John's request to be a plan to eliminate Mary's competition for the television.

A particularly interesting instance of persuasion occurs when the plan used to persuade the other planner is INFORM REASON. For example, consider the following story:

(25) John wanted to watch the football game but Mary wanted to watch the ballet. John told Mary that her mother got sick and was taken to the hospital.

Mary's mother is probably not sick. Instead, John is trying to cause Mary to have the goal of visiting her mother at the hospital. Being at the hospital would conflict with Mary's goal of watching the ballet. John is hoping that Mary will resolve this goal conflict in favor of being with her mother, and thus leave the television free for him to use. This technique of eliminating the opposition can be summarized in the following rule:

Rule S9:

If

1. a set of characters have competitive goals,

and

2. one character informs another character of a fact that causes that character to have a goal that conflicts with the goal of that character that was in competition,

then

that action is part of a plan to fulfill its actor's goal by eliminating the opposition.

For example, informing Mary that her mother was sick will probably give Mary the goal of visiting her. This goal conflicts with her goal of watching the ballet according to the rules for goal conflict detection discussed previously. Thus Rule S9 suggests that John informed Mary of this fact to make her abandon her plan involving use of the TV.

There is another rule that could be added about this situation. If a character uses INFORM-REASON for the purpose just described, then the information conveyed is probably not correct, unless the reader has prior knowledge that the reason is in fact true. That is, a person who states a reason for the competition to leave is probably telling a lie.

### 9.3.4.1 Complying with a threat. -

A technique similar to persuasion is applicable to goal competition situations in which one character threatens another. For example, consider the following story:

(26) Mary told John she would divorce him if he didn't give her more money.

Mary gives John a choice between two presumably undesirable alternatives: giving up Mary or giving up his money. John now has two preservation goals that are in conflict, that of preserving his money and of preserving his relationship with Mary. One plan for fulfilling a preservation goal specific to those caused by threats is to give in to the threat, abandoning the conflicting goal. I represent this situation by means of a planbox called COMPLY. The COMPLY planbox is a way to fulfill a goal caused by a another planner's THREATEN planbox by giving in to that planner's threat. For example, if story (26) were followed by

(27) John told Mary he would increase her allowance.

this sentence could be explained as an instance of the COMPLY plan for John's goal of preserving his relationship with Mary. This can be summarized by the following rule:

Rule P1:

If

a character threatened by another character performs the action requested in the threat,

then

this action is part of a plan to fulfill a preservation goal caused by the threatening character.

John telling Mary he will increase her allowance matches the request made of John by Mary. Rule P1 therefore states that John performed this action to fulfill the preservation goal of maintaining his relationship with Mary. The representation produced by this story is shown in Figure 10.

Figure 10
John tries to keep Mary by giving in to her demand

```
Goal                                Goal
 John preserve relationship          John preserve control
  with Mary                           of money
 Source: Preservation theme          Source: Preservation theme
  Initiator: (Mary's threat)          Initiator: (Mary's threat)
 Outcome: ?                           Outcome: Abandoned

        |
        |
        |

     Plan                      Action
      COMPLY        ----->    (John agrees to give Mary money)
       Planner     = John
       Requestee   = Mary                                |--> Mary
       Request     = John <=> ATRANS <--o-- money --|
                                                         |--> John
```

Figure 10 shows only John's goals. Mary's goal of getting more money should of course be included in the actual representation. Her goal gives rise to both of John's goals through her threatening action, which generated a preservation goal and caused John to have the above goals. Note that the outcome of John's goal of preserving his money shows that the goal has been abandoned. Using the COMPLY planbox in service of a preservation goal caused by a threat automatically implies that the conflicting goal is abandoned.

The outcome of John's other goal is still unknown. This is because the effectiveness of the COMPLY plan cannot be determined until the threatening character takes some action. After all, Mary may decide to leave John anyway.

The success of a COMPLY plan can be inferred by a reader according to the following rules:

Rule COMPLY1:

If

1. a character use the COMPLY plan to fend off a threat, and

2. the threatening character agrees not to perform the threatening action,

then

the COMPLY plan has succeeded in fulfilling the character's goal.

Rule COMPLY2:

If

1. a character use the COMPLY plan to fend off a threat, and

2. the threatening character undoes a precondition for the plan by which he would carry out his threat

then

the COMPLY plan has succeeded in fulfilling the character's goal.

Rule COMPLY3:

If

1. a character use the COMPLY plan to fend off a threat, and

2. the episode terminates

then

the COMPLY plan has succeeded in fulfilling the character's goal.

For example, sentence (27) could be followed by

(28) Mary agreed to stay.

or by

(29) Mary put down her suitcase.

Either of these sentences would imply the success of the COMPLY plan and the fulfillment of John's goal of preserving his relationship with Mary. Sentence (28) conforms to the conditions of Rule COMPLY1, since Mary is informing John that she will not make good her threat. Rule COMPLY2 can be used for sentence (29). Someone leaving a marriage relationship may move out, and take worldly possessions along. A suitcase can be used to carry these possessions, and to use a suitcase one must hold it. Thus putting down the suitcase undoes the precondition that the suitcase be held. Since this meets the requirements of Rule COMPLY2, a reader could infer that John has achieved his goal of maintaining his relationship to Mary.

If the story ended right after sentence (27), the reader would probably infer that Mary had been convinced to stay. This situation is handled by Rule COMPLY3.

Of course, a person may also used the standard set of anti-plans mentioned in the previous section to deal with preservation goals caused by threats. For example, in story (28)-(29), John could try to persuade

Mary to spend less money, or threaten to kill her if she left him.

### 9.4 Easing the competition

In each of the cases seen so far, each character whose goal competed with that of another character tried to fulfill his own goal at the expense of his competitors. An alternative approach to pursuing one's own goals is to try to ease the competition. For example, recall the following story:

(30) John wanted to watch the football game. Mary wanted to watch the Bolshoi ballet. John got the old black and white TV out of the attic.

Rather than risk a confrontation, John tried to ease the competition by acquiring an additional functional object. Since the competition between his goal and Mary's was based on their need to use the same functional object at the same time, John's action alleviates the competition by providing an additional resource.

As was the case for resolving goal conflict, easing goal competition depends upon the kind of goal competition present. Recall that in the previous chapter, three classes of goal competition were defined: Goals could compete because of a limitation of resources, because achieving the goals required the establishment of mutually exclusive states, or because the fulfillment of one character's goal caused another character to have a preservation goal. Easing goal competition is now discussed for each of these classes.

#### 9.4.1 Easing Competition Based on Resource Limitation

The previous chapter made a distinction between goal competition based on a limitation of a consumable susbstance and those based on a shortage of non-consumable functional objects. In this section these differences are reflected in the techniques by which each form of competition can be reduced.

For example, story (30) above is an instance of easing goal competition based on a shortage of a non-consumable functional object. In this story both John and Mary require use of the same television at the same time, so their goals are in competition. To relieve this competition, John acquired another television set. Since each planner could now pursue his own goal independently from the other, the goal competition is eased. This conforms to the following rule:

Rule S10:

If

1. a set of characters have goals that compete due to a shortage of a non-consumable functional object that the characters need to use simultaneously,

and

2. one character tries to acquire additional functional objects of the same kind,

then

that action is part of a plan to ease the goal competition.

Since shortages of a non-consumable causes competition only when several planners need to use the object concurrently, another way to ease the competition is to try to ease the time constraints on a character's plan. For example, consider the following story:

(31) John wanted to watch the football game, but Mary wanted to watch the Bolshoi ballet. John called up the ballet company and asked them to schedule their performance after the game.

Although John's plan is not very likely to succeed, it is a possible way to avoid the competition. If the ballet company agrees, the concurrent time restriction on Mary's and John's plans is lifted, and each of them can use the television independently. Thus story (31) could be understood with the aid of the following rule:

Rule S11:

If

1. a set of characters have goals that compete due to a shortage of a non-consumable functional object that the characters need to use simultaneously,

and

2. one character tries to change the time constraints on one of the characters' plans so that it need not be concurrent with the other character's plans,

then

that action is part of a plan to ease the goal competition.

Resource shortages that are caused by a lack of consumable objects can be eased somewhat differently. For example, consider

(32) John wanted to buy a stereo. His wife Mary wanted to buy a television, but they had only enough money to buy one. John decided to take a second job.

Here the strategy is to ease the competition by acquiring additional resources. Story (32) can be understood with the aid of the following rule:

Rule S12:

If

1. a set of characters have goals that compete due to a shortage of a consumable object that is needed by each character,

and

2. one character tries to acquire more of this consumable,

then

that action is part of a plan to ease the goal competition.

Rule S12 is applied as follows: Upon encountering the goal competition situation in story (32), the reader sets up a number of predictions, each corresponding to one of the above rules about insufficient consumable resources. Rule S12 is used to predict that one of the characters may try to get more money. When the reader learns that John decided to take a job, none of these predictions will be confirmed yet, so the input will be subjected to bottom-up processing. The explanation found for taking a second job is that it is a plan for the goal of getting money.

Each of the predictions is permitted to look at this inference to see if the inference mets its condition. Since the prediction created from Rule S12 is looking for an attempt to get more money, the prediction is confirmed, and the input is interpreted as a plan to ease the competition.

### 9.4.2 Easing Competition Based on Mutually Exclusive States

There are two ways states could exclude one another. The states could be logically exclusive or socially exclusive. If a goal competition situation is based on socially exclusive states, then it is possible to ease the competition by agreeing to violate social norms. For example, consider the following story:

(33) John wanted to marry Mary. So did Bill. They decided to all move into a house together.

The goal competition in story (33) arises because being married to two people violates a social rule. This competition can therefore be eased by breaking that rule and establishing both states anyway. This solution is not complete however. As is the case with goal conflict, violating a social rule usually has some undesirable consequences. Thus this solution may produce goal conflicts based on generating a preservation goal for all the characters involved. For example, John, Mary and Bill might be subjected to social ridicule for their behavior.

Thus the goal competition situations based on socially exclusive states can be understood with the aid of the following rule:

Rule S13:

If

a set of characters have goals that compete due to socially exclusive states,

then

that competition can be eased by violating the social rule causing the competition (and incurring goal conflicts for each character between his original goal and the preservation goals that would arise from violating the social rule).

Socially based goal competition can also be eased by changing one's society. For example, consider the following story:

(33) John wanted to marry Mary. So did Bill. They all decided to move to Tahiti.

The social rules in Tahiti presumably permit multiple marriages. By becoming part of this society, the social rule underlying the competition disappears. Thus we have the following rule:

Rule S14:

If

a set of characters have goals that compete due to socially exclusive states,

then

the planners can ease the competition by becoming part of a society in which the desired states are not socially exclusive.

If the goal competition is based on logically exclusive states, then the competition cannot be eased without one of the characters changing a plan for his goal.

### 9.4.3 Easing Competition Based on Generating a Preservation Goal

Goal competition situations based on generating a preservation goal can be eased if the character whose action causes the preservation state is willing to select an alternate plan. For example, consider the following cases:

> (34) John wanted to go to California with Mary, but Mary was afraid of flying. John agreed to drive there.
>
> (35) John asked Mary to stop smoking because the smoke annoyed him. Mary smoked her cigarette in the bathroom.
>
> (36) John wanted to go to the track, but his wife didn't want him to go because she thought gambling was immoral.
>
> (37) Johnny wanted an ice cream cone, but his mother thought it would spoil his appetite.

In story (34), John's plan for going to California causes goal competition because flying causes Mary to have the goal of easing her fear. John agreed to an alternate means of transportation, thus alleviating the problem. Mary alleviated the competition in story (35) by agreeing to smoke the cigarette where it would not annoy John. Stories (36) and (37) are not amenable to having their goal competitions eased since the plans for these goals do not have substitutes. That is, there is no way to enjoy gambling other than to gamble, and there is no way to enjoy eating an ice cream cone without eating one.

Thus easing goal competition based on generating a preservation goal can be done according to the following rule:

Rule S15:

If

a set of characters have goals that compete because the fulfillment of one character's goal would cause another character to have a preservation goal,

then

the competition can be eased if that planner is willing to change the plan for his goal to one that does not generate a preservation goal.

## 9.5 External competition removal

Goal competition can be eliminated by some action not taken by one of the story characters. This is the case in following stories:

(38) John wanted to watch the football game, but Mary wanted to watch the ballet. Mary put on channel 3. She found out that the ballet was postponed until later that day.

(39) John and Bill were stranded in the desert with only enough water for one of them to make it to the nearest town. Then they stumbled upon an oasis.

In story (38), John and Mary had competing goals because they needed to use the same object at the same time. However, when the ballet gets postponed, the competition vanishes because the time constraints on the characters' plans no longer force them to be done concurrently. In story (39), John and Bill had competing goals because of the lack of water. Finding a new supply of water alleviates the shortage of this resource, and therefore eases their goal competition. This removes a condition that caused the competition, enabling independent goal pursuit.

A reader can understand how the competition is eased in these stories using rules that are essentially the same as the rules presented in the previous sections of this chapter. That is, these stories are similar to those stories in which a character tried to change a time constraint or tried to obtain more of a consumable object.

Thus when a reader encounters an event following the occurrence of a goal competition situation in a story, the reader must check to see if that event eases the competition even if the actor of the event is a different character that one involved in the competition. The process of determining that the competition has been eased is not substantially changed. For example, Rule S11 above is as follows:

Rule S11:

If

1. a set of characters have goals that compete due to a shortage of a non-consumable functional object that the characters need to use simultaneously,

and

2. one character tries to change the time constraints on one of the characters' plans so that it need not be concurrent with the other character's plans,

then

that action is part of a plan to ease the goal competition.

To handle the case where the competition eases itself, this rule can be altered to become:

Rule S11':

If

a set of characters have goals that compete due to a shortage of a non-consumable functional object that the characters need to use simultaneously,

then

changing one of the time constraints on one of the characters' plans so that it need not be concurrent with the other characters' plans eases the goal competition.

In addition, this rule must be supplimented with the following rule:

Rule TRYCHANGE:

If

some condition contributes to goal competition among some characters,

then

one of the characters may try to change this condition to ease the competition.

Rule TRYCHANGE is a general rule about trying to ease a competitive situation. Rule S11' states a way in which a condition can contribute to competition. The other rules of the previous section can be similarly modified to produce rules like S11' stating conditions that contribute to competition. These new rules will handle the situations in which the goal competition is resolved for the planner, and together with Rule TRYCHANGE, they explain the behavior of a character trying to ease the competition. This situation is similar to that discussed in Chapter 7 on Goal Conflict Resolutions.

## 9.6 Summary

There are several paths a story might follow once a goal competition situation has occurred. Characters may try to pursue their goals while ignoring the competing goals of others. Sometimes a character will try to undermine his opponent in order to achieve his own goal, or try to persuade his opponent to alter his goal or plan. In addition, a character can try to ease the goal competition in several ways. Lastly, the goal competition may fade upon some chance event or upon the actions of some character not involved in the competition.

To understand stories involving competing goals, a story reader must understand how each character is behaving with respect to the goal competition situation. The reader must be able to use knowledge about goal competition situations to identify the situation of which the given

story is an instance. The reader can thereby explain each character's actions and infer the effects of these actions on the outcome of the characters' goals.

The knowledge needed to understand these situations takes the form of a large number of rules. These fall into several categories. In situations in which each character pursues his goal independently of the other characters, most of the inferences concern the outcome of competing goals. There are three rules for this: Rule S1 states that one character may pursue his goal, and the others abandon theirs. Rule S2 applies when a shared consumable resource is involved, and says that if enough planners pursue their respective goals, then all of them may fail. Rule S3 says that if the goal competition is based on mutually exclusive states, then each planner pursuing his own goal may lead to one victor while the other planners fail.

The next major category of rules concerns anti-planning. Most of these rules are needed to determine that an action is part of an anti-plan. Rules S4-S6 are used to detect anti-plans that undermine the opponent's plan. They recognize efforts at physically disabling an opponent, physically interfering with his plan, and establishing a state that mutually excludes a precondition for an opponent's plan, respectively.

Anti-planning made use of the following anti-planboxes:

1. OVERPOWER - Used to physically eliminate an opponent. Applicable in any anti-planning situation.

2. UNDO-PRECONDITION - Sabotage an opponent by undoing a precondition for one of his plans. Useful in any anti-planning situation.

3. AVOID-DETECTION - Perform a plan so as not be observed. Useful for anti-planning in situations involving anticipated preservation goals.

4. DISTRACT - Getting an opponent to focus his attention else while a plan is executed. Also applicable to situations involving anticipated preservation goals.

5. COMPLY - Giving in to an opponent in hopes of fulfilling a preservation goal. Useful in response to threats.

Next, there are rules to recognize defensive plans. Rules PHYS1 through PHYS4 describe how to avoid physical threats and dangers. Rules ANTICIPATE1 through ANTICIPATE4 detect efforts to prevent giving someone else a goal that is antithetical to one's own. These can identify plans to avoid detection, to avoid unleashing natural forces, to distract an opponent, or to reduce an opponent's physical abilities.

Another defensive rule is Rule P1, which is specific to defending against threats. This rule states that an threat can be avoided by complying with the threatener. Rules COMPLY1 through COMPLY3 are then

needed to determine if compliance actually nullified the threat.

The last set of rules for anti-planning involve persuading an opponent. Rule S7 detects attempts to persuade an opponent not to pursue a goal; Rule S8 detects attempts to persuade an opponent to pursue another goal; amd Rule S9 detects attempts to delude an opponent into pursuing a diversionary goal.

The final set of rules concerned easing the competition between planners. Rule S10 through S12 recognize efforts to change a condition giving rise to the competition based on resource limitation. The rules handle obtaining more functional objects or undoing a time restriction. Rules S13 through S14 detect plans aimed at easing competition based on socially exclusive states, and Rule S15 detects efforts to ease competition by selecting a different plan.

Finally, a modified subset of these rules is needed to understand situations in which the competition is removed by an external event.

While there are a fair number of these rules, only a few of them at most are applicable to similar situations. For example, the rules that give rise to predictions generally are dependent upon the particular type of goal competition that is present in the story. Thus after the type of goal competition has been determined, only a few predictions need be made for subsequent processing.

# CHAPTER 10

## GOAL CONCORD

People often recognize a common interest across their goals, and may work together toward the same end. A situation in which a set of characters recognize that their goals are related and work together towards them is called a goal concord situation. A story understander needs to recognize that the goals of several characters concur with one another in order to understand those characters' actions with respect to each other's goals.

To detect concordant goals, goal concord may be broken down into a number of cases. Goal concord may occur if one character intentionally causes another to have a goal that concurs with his. There is goal concord based on group goals, in which two characters have concordant goals because they are both members of some group. Two parties may recognize a common interest and form an alliance to address their goals together. There is also thematic goal concord, in which the existence of a thematic relationship between characters cause one to take on the other's goals.

Goal concord leads to a number of story situations that can only be understood in terms of the concordant nature of the characters' goals. For exanple, characters may pool their resources or abilities together in a common plan for their goals; they may divide responsibility for the execution of a plan for their goals among them, and one character may anti-plan for another in the case where a concordant goal also competes with a goal of another character.

## 10.1 Introduction

The world is not really as brutal a place as the last few chapter would make it seem. People often help each other with their goals as well as compete with one another. Story characters may share common goals, or have goals whose plans aid reather that hinder one another. To understand situations in which characters' goals may affect each other positively, the story reader cannot follow each independently. As was the case for goal conflict and goal competition, the reader must understand the interrelation of the characters' goals in order to explain each character's behavior.

For example, consider the following stories:

(1) John hated Fred. He hired Bill as a hit man, and gave him a gun.

(2) John hated Fred. He hired Bill as a hit man. Then John went over to Fred and gave him a gun.

(3) John and Bill were partners playing golf. Bill hit a shot into the rough. John sneakily moved the ball into a better position.

(4) John and Bill were playing golf. Bill hit a shot into the rough. John sneakily moved the ball into a better position.

In story (1), John employs Bill to perform a task for him. By hiring Bill, John has assured that both he and Bill have the goal of Fred being dead. If a person's goal is identical to that of another person, then one person may help another person to achieve his goal. One way to help someone achieve a goal is to fulfill a precondition for a plan for that goal. Using a gun is a way to reduce someone's physical state, and a precondition for using a functional object is to have one. Thus a reader of story (1) can explain John's giving a gun to Bill by inferring that John and Bill shared a common goal, and that John was fulfilling a precondition for the plan Bill was likely to use.

Understanding this story required that the reader infer the commonality of goals between John and Bill. In contrast, most readers find John's behavior in story (2) difficult to explain: there is no commonality of goals between John and Fred, so the reason for giving him a gun is unclear. A reader might infer that John had some extraordinary sense of fair play, and was giving Fred a chance to protect himself. But this explanation is different from the one reached in story (1) when the goals of the characters were in unison.

Stories (3) and (4) contain similar situations. In story (3), John and Bill are supportive of each other's goals by virtue of being on the same team. A reader can explain John's cheating to help Bill because each character wants the other to fulfill his respective goal. In story (4), however, John and Bill should be involved in a goal competition relationship. Since John appears to be helping rather than hindering

Bill, the reader should find John's action difficult to explain. In fact, some readers of stories (2) and (4) reported that they initially mis-read the stories. These readers confused Bill with Fred in story (2), and thought that the noun phrase "the ball" in story (4) referred to John's ball rather than Bill's.

Thus the situations in these stories are so strongly suggestive of cooperation rather than competition that readers chose to interpret the stories in a way that did not conform to the actual text. Upon re-examination of the stories, some readers tried to explain a character's behavior by re-examining his goals. For example, a reader might infer that John had been paid off to lose to Bill in story (4). With this assumption, John's goal would be the same as Bill's, and John's behavior could be readily explained as it was in story (3).

### 10.1.1 Goal Concord

The relationship between John's goal and Bill's goal in stories (1) and (3) is called goal concord. Goal concord is a situation is which two or more characters are supportive of each other's goals. For example, in story (1), John has enlisted Bill to do something for him. Both characters then had the same goal of Fred being dead. In story (3), John and Bill have a mutual interest in each other's goal because each character is dependent upon the other's goal being fulfilled in order to win.

In story understanding, a reader must detect goal concord in order to explain a character's behavior with respect to the plans and goals of the other story characters. For example, suppose that after reading story (1), a reader were asked the question

Q1) Why did John give Bill a gun?

To compute the answer

A1) He wanted him to use it to shoot John.

the reader would have had to infer that John was fulfilling a precondition for a plan of another character with whom he shared a common goal. When a character performs an action that fulfills a precondition of a plan for another character's goal, and that goal concurs with a goal of the first character, then a good answer to why that action took place is to state that the first character wanted the second character to perform his plan. Since a plan associated with having a gun is shooting someone, the reader could respond with answer A1 above.

Answering question Q1 required the reader to have determined that John and Bill have concordant goals. Then the reader's knowledge about goal concord could be used to explain why John performed an action helpful to Bill. Thus a story understander must be able to detect concordant goals and infer their effects on each character's plans. In the next section, I discuss the various kinds of concord that can exist

between goals, and give algorithms for detecting them. The following section is concerned with understanding a story once concordant goals have been spotted.

## 10.2 Detecting Goal Concord

To understand a story in which goal concord explains a character's behavior, a reader must first recognize that the character's goals concurs with someone else's. Recognizing goal concord requires the reader to have knowledge about how goals can be in concord. In this section I propose a classification of goal concord situations based the kinds of knowledge required to spot each situation. An understander supplied with this knowledge could use it to determine if goal concord exists between the goals it is examining.

### 10.2.1 Kinds of Goal Concord

I found the following classes of goal concord to be useful:

1. Induced goal concord

   One character may try to get another to have the same goal as he has. Employing someone as an agent is an instance of induced goal concord.

2. Group goals

   Two characters may have concordant goals because they are both members of some group. For example, teammates have goals that concur by virtue of their belonging to the same team.

3. Alliances

   Two characters who happen to have similar goals may join forces to help each other out. Allies in war is an example of an alliance between to countries having similar goals.

4. Thematic goal concord

   A character may be in a thematic relationship with another character that produces concordant goals. For example, if one person loves another, then that person may take on the goals that the loved one develops.

### 10.2.2 Induced Goal Concord

A person may employ another person as an agent. When a person persuades someone else to do his bidding, then the agent has a goal that concurs with the planner's. For example, story (1) above was an instance of this form of goal concord:

> (1) John hated Fred. He hired Bill as a hit man, and gave him a gun.

John employs Bill as an agent against Fred, causing Bill to have the same goal as he has. The reader could then understand why John gave Bill a gun: Bill's inferred goal of killing Fred concurs with John's goal of Fred being dead. Thus John helped Bill to fulfill one the preconditions for the plan for his goal.

Identifying this form of concord is essentially the process of detecting someone employing an agent. Usually, this consists of an application of the persuade package to the intended agent. For example, John persuaded Bill to become his agent in (1) by offering him money. This can be summarized in the following rule:

Rule GCONC1:
if
a planner persuades another planner to do some plan for him,
then
the planner and his agent have concordant goals.

Note that Rule GCONC1 separates agency from simple threats and bargains. In the case of agency, the request made of the agent must be one that the planner can perform, but doesn't wish to. For example, threatening someone to give you something does not enlist that person as an agent because the planner could not himself execute the plan of giving himself someone else's property.

### 10.2.3 Group goals

Groups often have common goals. When something is in the interest of the group as a whole, each member of the group may have a concordant goal. For example, consider story (3) above:

> (3) John and Bill were partners playing golf. Bill hit a shot into the rough. John sneakily moved the ball into a better position.

John and Bill are a members of the same team. A team is a group that is defined by its common goals. That is, a team is a group of people explicitly working toward a particular goal, usually a goal that competes against that of another team or individual. In story (3), John and Bill constitute a team with respect to their golf game. Thus in the context of that game, a reader can infer that their goals will concur.

This inference can be used to explain why John cheated on Bill's behalf: A person whose goal concurs with another person's may take some action to help that other person achieve his goal.

A reader usually detects this form of goal concord by being told that some characters are members of a group that has some common goals. This is summarized by the rule:

Rule GCONC2:

If

1. two planners are members of the same group,

and

2. that group has some common goals,

and

3. one planner has a goal that is instrumental to the goal of the group,

then

the other planner will have a goal that concurs with the first planner's goal.

Groups that can give rise to concordant goals include most established institutions and political entities, sports teams, and work units (e.g., programming teams, construction crews). Rule GCONC2 states that if the reader learns that two people of members of such a group, then they will have concordant goals with respect to the goals of that group. For example,

(5) The Democratic governor was having trouble winning the election, so the President flew in to campaign on his behalf.

Here the President and the governor are presumably members of the same party. Since a political party has as its goal the election of candidates from within its ranks, the President has a goal that concurs with that of the governor. Thus an understander can infer that the President might be helping the governor out because they are both members of the same political group.

### 10.2.4 Alliances

People who learn that they share a common interest may decide to pool their efforts together and form an alliance. An alliance is a situation in which two characters explicitly decide to unite on a common goal. For example, consider the following story:

(6) John and Bill both hated Fred. They decided to wait for him after school.

In story (6), John and Bill apparently discover that they have a common goal, and decide to pursue it together. Both John and Bill wish to do Fred harm, and presumably wait together to gang up on him. This is in conformance with the rule:

Rule GCONC3:

If

1. two planners have goals whose goal states are the same,

and

2. decide to do an action together which is part of a plan for each character's goal,

then

the two planners have formed an alliance based on their concordant goals.

For example, in story (6), a reader can infer that if a person hates someone, he may want to do harm to that person. John and Bill may both have this goal, and take an action together to fulfill it. Thus according to Rule GCONC3, John and Bill have formed an alliance with respect to their common goal of harming Fred.

Alliances are often formed in response to commonly anticipated preservation goals, particularly among political entities. For example, consider the following story:

(5) The United States and China were afraid of the Soviet Union, so they signed a mutual defense pact.

Here the US and China each has an anticipated preservation goal of protecting itself from the Soviet Union. Thus they decided to take a common action against the Soviet Union if the need should arise. Rule GCONC3 therefore states that the US and China have formed an alliance based on their mutual goals of preserving themselves from Soviet aggression. The political aspect of alliances is discussed in more detail in Carbonell (1978).

### 10.2.5 Thematic goal concord

A reader is often required to infer goal concord based on certain thematic relationships that exist between story characters. For example, consider the following story:

(7) John loved Mary. John knew that Mary wanted a new bracelet, so he gave Mary the money for one.

Love is an instance of an attitudinal theme that gives rise to goals with respect to the attitude object. One rule about love is that the person who has the attitude usually will adopt the goals of the attitude object. For example, if John loves Mary, and Mary wants something, then John may have the goal of Mary getting what she wants.

In story (7), Mary has the goal of have a new bracelet, and therefore John will also want her to have it. Since their goals concur, a reader can understand why John gave Mary the money she needed. He was helping her to achieve a goal that he also had for her. This is expressed by the following rule:

Rule GCONC4:

If

1. a character has a thematic relationship toward another character that makes him adopt the goals of that character,

and

2. if that character has a goal,

then

the first character will have a goal that concurs with the goal of the later character.

Anther example of thematic relationships involving goal adoption is being responsible for someone's well-being. For example, a peron entrusted with the care of a child will adopt certain of the child's goals, such as protecting the child from harm. These particular thematic relationships are discussed in Schank et al., (1978).

## 10.3 Goal Concord Situations

Goal concord situations are about the ways in which one character can aid a character with whom he has a concordant goal. These situations fall into three classes based upon how characters can help one another:

1. Pooling resources and abilities
2. Pursuing another character's subgoals.
3. Pursuing another character's preservation goals.

Each type of situation is now discussed in turn.

### 10.3.1 Pooling resources and abilities

Two characters with concordant goals may have trouble achieving their goals independently because of a lack of a resource or an ability. By merging resources together, the characters may be able to execute a plan for their goals that they could not execute independently. For example, consider story (6) above:

> (6) John and Bill both hated Fred. They decided to wait for him after school.

In story (6), John and Bill presumably are in concord to beat up Fred. They are probably going to use the OVERPOWER planbox to do this, and one of the preconditions for using OVERPOWER is that the planner is more powerful than the victim. By joining together, John and Bill are able to apply their combined strength against Fred, thus increasing the chances of a favorable outcome for their plan. Story (6) therefore conforms to the following rule:

Rule GCONC5:

If

1. two characters have goals that are in concord,

and

2. a plan for each goal requires a resource or an ability possessed by both characters,

then

the characters may apply their resources or abilities jointly to the execution of the one plan to achieve both their goals.

## 10.4 Pursuing parts of a joint plan

Characters with concordant goals can also work together on various parts of a joint plan. For example, consider the following story:

> (8) John and Mary were hungry. Mary went to the grocery store and John cooked dinner.

Here John and Mary have similar goals, and ally their forces together to fulfill both goals. Mary fulfills one precondition for their plan by getting some food, and John fulfills another by preparing it. This is in accordance with the following rule:

Rule GCONC6:

If

1. two characters have goals that are in concord,

and

2. a plan for each goal has several preconditions,

then

the characters can achieve both their goals by individually fulfilling some of the preconditions for the plan.

10.4.1 Concord in goal competition situations.

An interesting instance of goal concord occurs when two characters have concordant goals, and one of the goals competes with that of another character. For example, consider the following story:

(9) John and Mary wanted to get the treasure, but the way was blocked by a ferocious dragon. Mary distracted the dragon while John ran and got the treasure.

In story (9), John and Mary have the concordant goals of possessing the treasure, but getting the treasure is likely to cause the dragon to harm them. This is an instance of goal competition caused by generating a preservation goal, and was discussed in Chapters 8 and 9.

One of the ways of dealing with such a goal was to use the DISTRACT planbox, which directed the attention of the potential adversary away from the planner's action. Because there are two characters with concordant goals in story (9), one of them can perform the DISTRACT planbox, while the other character pursues the original goal. This is an instance of the following general rule involving goal concord in competition situations:

Rule GCONC7:

If

1. two planners have goals that are in concord,

and

2. one of the goals is in competition with the goal of another character,

then

one planner can anti-plan against his competitor while the first planner pursues the original goal.

Rule GCONC7 is applicable to most goal competition situations. For example, is John is in a race against Fred, and Bill has his money bet on John, then Bill might take some action against Fred to help John win. Thus a person whose goal concurs with someone in a goal competition situation may take any action that the person in competition might himself take.

## 10.5 Summary

Characters often join together when they recognize a common interest. A story understander needs to recognize that the goals of several characters concur with one another in order to understand those characters' actions with respect to each other's goals. In order to detect concordant goals, it is useful to define several types of goal concord: There is induced goal concord, in which a character causes another character to have a goal that concurs with his; there is concordance based on group goals, in which two characters have concordant goals because they are both members of some group; goals can concur when two otherwise unrelated parties make an alliance toward a common end; and there is thematic goal concord, in which the existence of a thematic relationship between characters causes one to take on the other's goals.

Goal concord leads to a number of story situations. These include situations in which characters with concordant goals pool their resources or abilities together in a common plan for their goals, situations in which the characters divide responsibility for the execution of a plan for their goals among them, and situations in which the characters with concordant goals are also in goal competition. A story reader needs knowledge about each of these situations in order to explain the behavior of the characters that participate in them.

# CHAPTER 11

# OBJECTS

A story understander often requires knowledge about objects to explain the behavior of a story's characters. Knowledge about objects is needed for the understander to infer a character's plan, to detect resource shortages that may cause goal conflict and goal competition, and to infer that a character is trying to subsume a goal.

To understand stories in which objects play a role, several kinds of knowledge about objects is required. One kind of knowledge needed to understand stories involving objects is the <u>function</u> of an object. The function of an object is a description of the plans for which that object is designed. For example, the function of a gun is to play a role in a plan for overpowering or threatening someone. Knowledge about the function of an object is useful for inferring the plan of someone who has the goal of possessing an object.

In addition to knowing what plans an object can be used for, a reader must know how an object is affected by use in a plan. Some objects are <u>consumable</u> in that their use in a plan precludes their use again by the planner. For example, food is a consumable object because after it is used to satisfy one's hunger, it cannot be used again for this purpose. In contrast, a hammer is an instance of a <u>non-consumable</u> object. That is, a hammer can be used repeatedly by the same planner to perform the same function.

The notion of a consumable versus non-consumable object is important in story understanding because these objects act differently with respect to goal competition, goal conflict and goal subsumption situations. For example, ownership of a non-consumable object can subsume recurring goals for which that object is applicable. Possession of a consumable can subsume only as many goals as the amount of the consumable allows.

To distinguish consumable from non-consumable objects, the reader must know how an object is affected by use in a plan, and what the requirements of that plan on that object are. There are three ways in

which the changes to an object and the requirements of a plan can cause an object to be consumed. The object can be consumed physically, as food is when it is eaten. The object can be consumed socially, as money is when it is transferred to another planner. And the object can be consumed by continuous enablement, as a thumbtack is when it is used to continuously support another object.

The notion of the function of an object is an important part of the definition of an object. However, most attempts at finding semantic primitives for nouns have ignored this aspect of their definition. By including the notion of function in the definition of certain nouns, some of the difficulty of representing the meaning of these words is eliminated.

## 11.1 Introduction

Many of the inference rules given in the previous chapters referenced knowledge about objects. When a character in a story has some dealings with an object, information about that object may be needed to determine what plan the character has in mind, and what the consequences of performing that plan are to the planner. For example, consider the following stories:

(1) Willa was hungry. She picked up the Michelin guide.

(2) John wanted a new stereo and a new television, but he had only enough money for one.

(3) John got a job in Glastonbury. He decided to buy a car.

In story (1), the reader should infer that Willa is using the Michelin guide to find out where a restaurant is. To explain why Willa picked up the the Michelin guide, the reader must know that books normally have functions associated with them. One of the functions of a book is a source of information, and the Michelin guide happens to be a source of information about restaurants. Since knowing the location of a restaurant is a precondition for a plan for going there, the reader can use this knowledge about the purpose of a book to infer that Willa was going to eat at a restaurant.

Story (2) is an instance of a goal conflict based on the lack of a consumable resource. To recognize that John's goals conflicted, the reader must know that money is a consumable object that cannot be re-used for each of John's goals. Money is consumable because using it requires that the money be transferred to another person. In general, the reader needs to know which objects are consumable in order to detect goal conflicts based on a shortage of a consumable resource.

In story (3) John's new job entails a recurring goal of being in Glastonbury every day. John has decided to subsume this goal by acquiring ownership of a car. Owning a car subsumes this goal because it eliminates the need for John to obtain control over a vehicle every time he needs to use it. For a reader to determine that John's owning a car subsumes his recurring "be at work" goal, the reader needs knowledge about the relation of objects to goal subsumption states.

This chapter's aim is to codify the knowledge about objects needed to understand objects in relation to plans. It is not intended to be an exhaustive treatment of the object knowledge needed by an understanding system. It should not be surprising, therefore, that I ignore much of the other knowledge about objects that may be referenced by natural language utterances. In particular, there most certainly is a large body of knowledge about the physical appearance of objects, the physical properties that objects display, and the environments in which objects occur, all of which may have to be accessible to an understander. Some of these aspects of objects are discussed in Lehnert (1978).

## 11.2 Function

The most fundamental type of information about objects that is needed for natural language understanding has to do with the function of an object. The function of an object is the role that an object plays in a particular plan. This information is used during understanding to determine why a person performed a particular action with an object, and to predict what that person is likely to do next.

For example, consider the story

(4) John wanted money. He got a gun and walked into a grocery store.

John is probably going to rob the grocery store. To make this inference, a reader must know that getting a gun is an indication that a person will use the gun. The understander must also know the purposes to which a gun can be put. Then when the reader sees the second sentence of (4), he can use this knowledge to infer that John was going to use the gun to threaten or overpower someone. Since threatening and overpowering are ways to get something from someone, the understander could infer that John was going to perform one of these plans in order to get some money away from someone.

To demonstrate that a reader does in fact make this inference at this point in the story, note what happens if (4) is followed by

(5) John told the owner he wanted his money.

Most readers interpret (5) as an instance of a threat directed by John toward the owner of the grocery store. The reader can make this inference because he has previously inferred that John was going to threaten or overpower someone to get some money. When a plan for a

character has been inferred, the understander always tries to interpret an action by that character as a step in that plan. One of the steps in the plan for threatening someone is to inform that person of what is demanded of him. Thus the presence of an inferred THREATEN plan would cause the owner to interpret (5) as John threatening to shoot the owner with his gun if the owner doesn't give John the money.

Had this THREATEN plan not been inferred, a quite different interpretation of (5) might be found. For example, suppose (5) were preceded by

(6) John the Hostess salesman had sold a grocery store owner a lot of cupcakes on credit. John went down to the grocery store.

In this case, the reader would probably interpret (5) as John trying to collect the money that the store owner owed him by appealing to a social rule that people are obliged to pay their debts. This interpretation is supported because (6) suggests an entirely different plan than was suggested by (4). Thus the inference that John might threaten someone in story (4) was used by the understander to select among a number of plausible interpretations of the next event.

The inference that John might threaten or overpower someone in (4) is based on knowledge about the function of the object gun. Note what happens if the object in (4) is changed:

(7) John wanted money. He got his thermos bottle and walked into a grocery store. John told the owner that he wanted his money. The owner gave John the money and John left.

It is unclear why the owner gave John the money in this story. The story is confusing because the plan under which John is operating cannot be inferred. In story (4), the reader could use his knowledge about the function of a gun to connect up the events, but the function of a thermos bottle seems irrevelant to any of John's goals or actions.

To make the inference about John's plans in story (4), the reader must have the knowledge that a gun is a functional object. A functional object is an object whose memory representation contains a description of how that object is used in a plan. For example, a gun is a functional object because its memory representation contains the information in Figure 11.

The memory representation of a functional object may describe a number of plans in which that object plays a role. In Figure 11, there are two functional descriptions under GUN. The first states that a gun may be used for in the plan THREATEN. In fact, this plan is a particular version of THREATEN in which the threat is already specified. This specification is shown by the constraint on the THREAT role of the plan. This constraint states that the threat to be used will be that of shooting the person by means of doing the instrumental script by which one fires a gun.

Figure 11
Memory Representation of the Object GUN

```
GUN

 FUNCTION:

   [ THREATEN(PLANNER, PERSUADEE, REQUEST, THREAT)
        CONSTRAINTS

          THREAT IS
            PLANNER <=> PROPEL <--O-- PERSUADEE
                             ^
                             | INST
            PLANNER <=> $GUN-SCRIPT <--O-- GUN

     OVERPOWER(PLANNER, VICTIM, FORCE)
        CONSTRAINTS

          FORCE IS
            PLANNER <=> PROPEL <--O-- PERSUADEE
                             ^
                             |INST
            PLANNER <=> $GUN-SCRIPT <--O-- GUN        ]
```

The second functional description under gun states that a gun may be used in the OVERPOWER planbox. In particular, it is a version of OVERPOWER in which the means of overpowering is constrained to be the instrumental gun script. Of course, there are other constraints on the above plans that are not shown. For example, the PLANNER, the PERSUADEE, and the VICTIM all must be people. These constraints are not included above because they are general to all forms of the THREAT and OVERPOWER planboxes.

The information represented in Figure 11 can be used by a story understander like PAM to make the inference required to understand story (4). When an understander sees that John got a gun, it looks up this definition of GUN and find that a gun has a number of functions. It then tries to determine which of these functions is applicable in the context of the story it is understanding. This is accomplished by checking the plans in the functional description under GUN against the goals encountered in the story. Since the understander knows that both THREATEN and OVERPOWER can be used for the goal DCONT, it infers that John is going to use one of these plans for his goal of getting some money.

The understander then waits to see which of these plans John's behavior conforms to. This is done by trying matching the next event against each plan. Since informing someone of what is desired of him is

part of the plan for THREATEN, but not for OVERPOWER, the reader can infer that John is using THREATEN. Moreover, since the particular version of THREATEN found under GUN states the means of the threat, the understander can use this knowledge to infer that John was threatening to shoot the owner if the owner didn't comply with his request.

After the THREATEN plan is confirmed, the knowledge organized under THREATEN can be used to interpret subsequent events. If the owner gives John the money, for example, the understander can infer that he did this to prevent John from carrying out his threat. A trace of PAM run on this story was given in Chapter 1.

### 11.2.1 Applying knowledge about functional objects.

PAM can process the above story because it knows what a gun is. When PAM encounters a functional object like gun, it uses the knowledge supplied by the functional definition of that object to explain a character's behavior. Some of the further implications of the functional definitions of words like "gun" are discussed at the end of this chapter.

To apply the knowledge of the function of a gun to understand story (4), the reader needed to know that John intended to use the gun. This inference can be made by a general inference rule about functional objects, namely:

Rule 01:

If
    a person wants to control a functional object,
then
    he wants to do one of the plans mentioned in the functional description of the object.

In story (4), a reader can infer from John getting a gun that John wanted to have a gun. Then the above rule can be used to infer that John wanted to THREATEN or OVERPOWER somebody.

Note that this inference rule is based on a person having a DCONT goal, rather than the action of getting the object. This generalization allows the understander to make the same inference in a greater number of cases. For example, suppose that instead of "John got a gun", the understander were told "John asked his wife where his gun was". Using rules about asking, the understander could infer that John wanted to have a gun, and then invoke the above rule to predict John's next plan. No additional rule about functional objects would be needed for this example.

Some other examples of functional descriptions of objects are:

```
COFFEE POT
  FUNCTION:
    [ MAKE-COFFEE-SCRIPT(PLANNER,COFFEE,WATER,COFFEE POT) ]
```

That is, a coffee pot is used to make coffee. To use it, one executes a particular script. In this case, Rule 01 would let the understander infer that someone was going to make coffee if that person had the goal of having a coffee pot.

```
CAR
  FUNCTION:
    [ USE-VEHICLE(PLANNER,VEHICLE,LOCATION,FUEL)
        CONSTRAINTS:
          VEHICLE IS CAR                              ]
```

This describes that a car can be used to move oneself to a new location by driving there.

```
MONEY

  FUNCTION:

    [ BARGAIN(PLANNER, PERSUADEE, REQUEST, OFFER)

        CONSTRAINTS:
                                                  |-> PERSUADEE
   OFFER IS  PLANNER <=> ATRANS <--O-- MONEY -|
                                                  |-< PLANNER  ]
```

The functional description of money encodes the information that money can be used to make a bargain by giving the money to someone in exchange for that person complying with a request.

```
THERMOS

  FUNCTION:

    [ USE-CONTAINER(PLANNER, CONTAINER,
                    CONTAINED-SUBSTANCE,PROPERTY)

        CONSTRAINTS:

            CONTAINER IS THERMOS
            CONTAINED-SUBSTANCE IS LIQUID
            PROPERTY IS TEMPERATURE(LIQUID)     ]
```

A thermos is an object used to contain a liquid at a particular

temperature. The above definition of thermos states that a thermos can be used as the container in the USE-CONTAINER plan to contain a substance that is a liquid, preserving its temperature.

### 11.2.2 Computer Example

The following is a trace of PAM run on the Michelin Guide story, which was discussed previously in Chapter 3. To understand this story, PAM draws on the functional description of the Michelin Guide. This description states that the Michelin Guide is a source of information about where to find restaurants. PAM uses this knowledge to infer that Willa was going to go to a restaurant.

```
[PHOTO:  Recording initiated  Sat 12-Aug-78 12:09PM]

*(UNDERSTAND CD5A)

THE STORY IS

 WILLA WAS HUNGRY.
 SHE PICKED UP THE MICHELIN GUIDE
 AND GOT INTO HER CAR.
```

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(WILLA WAS HUNGRY) | |
| CONCEPTUALIZATION IS:<br>((ACTOR HUMO IS (*HUNGER* VAL (5.)))<br>TIME (FORM6)) | |
| NOT A PREDICTED INPUT | |
| BEGIN SEARCH FOR EXPLANATION | |
| TESTING EXPLANATION OFFERED BY<br>SHUNGER-EPISODE-REQ | |
| EXPLANATION IS GOAL:<br>(*SHUNGER* PLANNER HUMO)<br>WITH SOURCE (*SATISFY-DRIVE*) | PAM recognizes the input as a Satisfy-hunger goal, and from the hunger drive. |
| EXPLANATION CONFIRMS PREDICTION<br>INIT-REQ | |

FOUND EXPLANATION SEQUENCE:
SHUNGER-EPISODE-REQ

*** ADDING TO STORY REPRESENTATION:

LOADING PREDICTION
SUITABLE-PLAN-REQ

PAM predicts that Willa will pursue some plan for this goal.

INFERRED GOAL: (*SHUNGER* PLANNER HUMO)

NEXT INPUT IS:
(SHE PICKED UP THE MICHELIN GUIDE)

CONCEPTUALIZATION IS:
((ACTOR HUMO <=> (*GRASP*) OBJECT
PHYS1 TO (*HAND* PART HUMO))
TIME (FORM11))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
TAKE-REQ

EXPLANATION IS
PLAN: (*PB-TAKE* PLANNER HUMO
OBJECT PHYS1)

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
TAKE-GOAL-REQ

EXPLANATION IS GOAL:
(*DCONT* PLANNER HUMO OBJECT PHYS1
OWNER HUMO RECIPIENT HUMO)
PLAN: (*PB-TAKE* PLANNER HUMO
OBJECT PHYS1)

PAM infers that Willa wanted to have the Guide.

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

```
TESTING EXPLANATION OFFERED BY
USE-BOOK-REQ

EXPLANATION IS
 PLAN: (*PB-READ* PLANNER HUMO
OBJECT PHYS1)
```

Now PAM uses its knowledge about the function of a book to infer that Willa wanted to read the Guide. This is an implementation of Rule 01, which states that a person who wants to have a functional object wants to use it.

```
NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
READ-GOAL-REQ

EXPLANATION IS GOAL:
(*DKNOW* PLANNER HUMO RECIPIENT HUMO
FACT ((ACTOR ORGO IS (*PROX* PART
(*UNSPEC* CLASS (*LOCATION*))))))
 PLAN: (*PB-READ* PLANNER HUMO
OBJECT PHYS1)
```

The goal behind reading is either to find something out or to enjoy oneself. Since the Michelin Guide is described as a source of information about the location of restaurants, PAM infers that Willa wanted to know where to find one.

```
NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
DKNOW-DPROX-REQ

EXPLANATION IS GOAL:
(*DPROX* PLANNER HUMO OBJECT HUMO
 LOCATION (*PROX* PART ORGO))
```

Since knowing the location of a place is often instrumental to going there, PAM infers that Willa wanted to be at a restaurant.

```
NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
USE-RESTAURANT-REQ

EXPLANATION IS
 PLAN: ($RESTAURANT PLANNER HUMO
RESTAURANT ORGO)
```

Since doing the restaurant script can be used to satisfy one's hunger, the prediction that Willa would execute a plan for this goal is confirmed.

```
EXPLANATION CONFIRMS PREDICTION
SUITABLE-PLAN-REQ
```

```
FOUND EXPLANATION SEQUENCE:

TAKE-REQ -> TAKE-GOAL-REQ ->
USE-BOOK-REQ -> READ-GOAL-REQ
 -> DKNOW-DPROX-REQ ->
USE-RESTAURANT-REQ


*** ADDING TO STORY REPRESENTATION:

 INFERRED GOAL: (*DCONT* PLANNER
HUMO OBJECT PHYS1 OWNER (NIL)
RECIPIENT HUMO)
```

These inferences are added to the story representation.

```
 OUTCOME OF GOAL: (*DCONT* PLANNER
HUMO OBJECT PHYS1 OWNER HUMO
RECIPIENT HUMO) IS (*SUCCEED*)

 INFERRED GOAL: (*DKNOW* PLANNER
HUMO RECIPIENT HUMO FACT ((ACTOR
ORGO IS (*PROX* PART (*UNSPEC* CLASS
(*LOCATION*))))))

 INFERRED GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*PROX*
PART ORGO)

LOADING PREDICTION
SUITABLE-PLAN-REQ
```

PAM predicts a plan for going to the restaurant,

```
INFERRED
 PLAN: ($RESTAURANT PLANNER HUMO
RESTAURANT ORGO)

LOADING PREDICTION
DO-$REST-REQ
```

and also predicts the execution of the restaurant script.

```
INPUT CONFIRMS PREDICTION
SUITABLE-PLAN-REQ


NEXT INPUT IS:
((ACTOR HUMO <=> (*PTRANS*) OBJECT
  HUMO TO (*INSIDE* PART PHYSO)
  INST (NIL))
 TIME (FORM43))

NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION
```

TESTING EXPLANATION OFFERED BY
WALK-GOAL-EPISODE-REQ

EXPLANATION IS GOAL:
(*DPROX* PLANNER HUMO OBJECT HUMO
LOCATION (*INSIDE* PART PHYSO))
PLAN: (*PB-WALK* PLANNER HUMO
LOCATION (*INSIDE* PART PHYSO))

PAM infers that Willa wanted to be inside her car.

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH

TESTING EXPLANATION OFFERED BY
USE-VEHICLE-REQ

EXPLANATION IS
PLAN: (*PB-USE-VEHICLE* PLANNER
HUMO LOCATION (*PROX* PART (NIL)))

Being inside a vehicle is usually instrumental to using it, so PAM infers that Willa will drive somewhere.

***** ADDITIONAL INFORMATION FOUND
ABOUT PLAN FOR
GOAL: (*DPROX* PLANNER HUMO OBJECT
HUMO LOCATION (*PROX* PART ORGO))

Since this is a plan for changing one's location, the prediction that Willa will execute such a plan is confirmed.

FOUND EXPLANATION SEQUENCE:

WALK-GOAL-EPISODE-REQ ->
USE-VEHICLE-REQ

*** ADDING TO STORY REPRESENTATION:

PLAN: (*PB-WALK* PLANNER HUMO
LOCATION (*INSIDE* PART PHYSO))

The inferences are added to the representation.

INFERRED GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*INSIDE*
PART PHYSO))

OUTCOME OF GOAL: (*DPROX* PLANNER
HUMO OBJECT HUMO LOCATION (*INSIDE*
PART PHYSO)) IS (*SUCCEED*)

PLAN: (*PB-USE-VEHICLE* PLANNER
HUMO LOCATION (*PROX* PART (NIL)))

```
LOADING PREDICTION               |
DO-USE-VEHICLE-PB-REQ            |
                                 |
                                 |
                                 |
FINISHED UNDERSTANDING           |
PHASE                            |
```

---

```
**********************************************************************
*                                                                    *
*  The inferences made during understanding are elicited             *
* while answering questions and paraphrasing the story.              *
*                                                                    *
**********************************************************************
```

QUESTION: Q1

Why did Willa pick up the Michelin Guide?
Because Willa wanted to know where a restaurant was.

QUESTION: Q2

Why did Willa get into her car?
Because Willa wanted to get to a restaurant.

QUESTION: Q3

What were the consequences of Willa picking up the Michelin Guide?
This enabled Willa to read the Michelin Guide.

QUESTION: Q4

What were the consequences of Willa getting into her automobile?
This enabled Willa to drive somewhere.

QUESTION: Q5

How did Willa get into her car?
Willa walked to her automobile.

WHO SHOULD TELL THE STORY? *WILLA

I wanted to get something to eat, but I didn't know where a
restaurant was. So I picked up the Michelin Guide, and I
got into my car.

[PHOTO: Recording terminated Sat 12-Aug-78 12:10PM]

11.3 Consumption

In addition to knowing the function for which an object is normally used, a reader also must know what happens to an object when it is used. For example, consider the following stories:

(8) John was reading the newspaper. When he finished, he let Mary read it.

(9) John was eating an apple. When he finished, he let Mary eat it.

Story (9) seems unreasonable because a reader knows that one of the consequences of eating is that the object eaten is no longer fit to be eaten again. In contrast, story (8) is perfectly acceptable because the process of using a newspaper does not prevent the newspaper from being used repeatedly. Unlike a newspaper, an apple is consumable.

A consumable object is an object that is changed upon its use in a plan so that a planner cannot use it for that plan again. An apple is a consumable object because after it is eaten it cannot be eaten again. A newspaper is not consumable because it can be read over again repeatedly.

Of course, functional objects that are not consumable may change somewhat as they are used in a plan. For example, a car is a non-consumable functional object, even though continued use of a car will reduce its ability to function properly. The difference between a consumable object like food, and a non-consumable that is subject to wear has to do with the role that the change plays in the functioning of the plan. It was pointed out to me by Carbonell (Carbonell, 1978) that the change due to wear is usually thought of as being incidental to the use of the object for its intended purpose. In contrast, change in a consumable object is inherent in using it.

For example, it is conceivable (and desirable) that one could use a car for an indefinite period of time without incurring any wear on it, since the wear is an artifact of using the car, and not crucial for the car to perform well. However, it is not conceivable that a consumable object like food could be used repeatedly, since the change it incurs when it is used in a plan is inherent in the nature of that plan.

To distinguish consumable from non-consumable objects, a reader needs to know how the use of an object in a plan alters that object, and which alterations would prevent the planner from using the object again. There are actually several types of consumption that are worth distinguishing for language understanding. These distinctions are based on the way in which the object changes after it is used in a plan.

### 11.3.1 Physically consumable objects

The most obvious way an object can be consumed by a plan is by undergoing a physical change. The apple in story (8) is an example of such an object. After use, its physical characteristics changed so that it could not be used again for the same plan.

An object can be physically consumed if either

1. The object no longer exists after use in a plan.

2. The object has changed some characteristic so that it no longer meets a constraint imposed upon the object by a plan.

For example, dynamite is physically consumed in its use because it no longer exists after it is used in its usual plan. Food is physically consumable because after it is eaten, it is no longer edible, and hence violates a precondition on the object required for the plan of eating.

The fact that an object is consumable is computed by examining the consequences of using the object in the plan for which the object is designed. A consequence is something that happens to an object due to the role the object plays in a plan. Stored along with each plan in memory is a description of the consequences that the plan has upon the objects it uses.

For example, the functional description of FOOD can be represented as follows:

```
FOOD

 FUNCTION:

   [ EAT(PLANNER, FOOD) ]
```

That is, food is something used in the planbox EAT. EAT is a very simple plan that consists of the performing the primitive action INGEST on an object that has the characteristic of being food. EAT is different than the primitive action INGEST in that it organizes some knowledge that is not organized around INGEST. For example, the object must be food, it must be ingested through the mouth, it is often done when the actor is hungry, etc. EAT organizes specific knowledge about INGEST just as THREATEN organizes specific knowledge about MTRANS.

In particular, one of the inferences from EAT that is not true on INGEST in general is that the physical state of the food eaten deteriorates (This might not be true it the object INGESTed was a marble, for instance). The consequences of eating some food on that food is encoded as follows:

EAT(PLANNER,FOOD)

CONSEQUENCES:

[ FOOD TOWARD PHYS-STATE(-) ]

This represents the fact that stored under the plan EAT is the consequence that the food eaten is then in a different physical condition. In a story in which food were eaten, the reader could use this information to infer that the food is in a new physical condition. However, if a story were encountered in which a character used food to bargain with, say, rather than to eat, the understander should not infer that the food is no longer edible. By attaching these consequences to the plan rather than the object, the reader can determine the correct changes to make to the description of the object by determining the particular plan the character is using.

The point here is that consumable objects can only be defined with respect to a particular plan. Reading a newspaper does not destroy it, but using it to start a fire does. Spending money makes it inaccessible to the planner, but rolling up a dollar bill to snort cocaine does not.

Thus the fact that something is consumable is not part of the definition of that object, in the same way that the function of an object is. Rather, this is computed from the information in a plan about the plan's effect on an object. Thus if the object is used in some novel way, the ordinary consequence of using that object may not be inferred.

In PAM, consequences of using a plan are inferred by inference procedures attached to particular plans. For example, attached to the planbox EAT is a procedure that will alter the representation of the particular item of food used by the plan. When PAM determines that the plan has been carried out, it will run this procedure. The procedure will mark the item of food as being in a negative physical state. In effect, this procedure implements the rule that if a person eats some food, then that food is in an altered physical condition.

Marking the item as such is important in examples like these:

(10) After John finished eating an apple, he was still hungry. He went out and got another apple.

(11) After John finished hammering in a nail, he realized that he had to hammer in another one. He went out and got another hammer.

After learning that John ate an apple, the reader of (10) marks the representation of the apple to show that it is in a changed physical condition. The apple cannot be used for the plan EAT because a precondition of EAT would be violated, namely, that the FOOD role of the EAT planbox can be only be filled by an object that is a food and is in good physical condition. Thus the reader can determine why John got

another apple - he wanted to have an apple he could eat to satisfy his hunger, and the apple that he just ate no longer met the preconditions for the plan EAT. He was therefore trying to get another object that conforms to the precondition on EAT.

In contrast, (11) is difficult to explain because a hammer is not normally consumed when it is used to hit something. The representation for the hammer is not changed after this plan is used, and hence the hammer is still a suitable object for the plan. Thus the understander cannot find the same explanation for John's getting another hammer as could be found for him getting an apple, since John does not stop having such an object after it is used. Instead, the reader might try to find an alternative explanation more suitable for a non-consumable functional object. For example, the reader might assume that John broke the hammer, since this would disable its use in a subsequent plan and thus precipitate the need for an additional hammer.

To apply knowledge about the suitability of an object for a role in a plan, the understander must store along with each planbox the constraints on the objects that can fill each slot. The memory representation of EAT given above should therefore be modified to include this information:

```
EAT(PLANNER,FOOD)

 CONSEQUENCES:

   [ FOOD TOWARD PHYS-STATE(-) ]

 CONSTRAINTS:

   [ FOOD IS PHYS-STATE(+)
     FOOD IS POSS(PLANNER)
     FOOD IS food for PLANNER
     PLANNER IS HUMAN          ]
```

Here we have added the constraints that the role FOOD must be filled by something that is possessed by the planner, in a good physical state, and also happens to be food for the particular planner. This last constraint is necessary because the notion of "food" can only be defined in relation to the type of actor that is going to consume it. See Schank (1975a) for a discussion of this issue.

In sum, the representation of plans in the memory of an understander must contain two kinds of information about objects. First, the reader needs to know what using the plan does to the object. Second, it must know the requirements that the plan imposes upon the object. This information is used to keep track of the states of the objects used in plans, and to check if an object meets the requirements of a specific plan. The example just described shows how an understanding system can use this information to deal with objects that are physically consumed by use in a plan. This process can be summarized by the following rule:

Rule 02:

If

a person uses an object that is physically consumed in a plan

then

that object cannot be again used for that plan.

Applying this rule requires the understander to keep track of the consequences to the object caused by the use of that object in a particular plan. The condition of Rule 02 holds if the consequences of performing the plan causes the object to violate a precondition for use of the object in that plan. This requires the understander to re-compute a new description of the object, and then match this description against the restrictions imposed on the object by the particular plan.

The notion of consumability is particularly useful in understanding why a character needs a particular amount of a substance. For example, consider the stories

(12) John went to the store and bought two apples.

(13) John went to the store and bought two screwdrivers.

In (12) a reader can infer that John might eat both apples. That is, apples are physically consumable objects, and therefore if John wanted to eat two of them, he would have to have two of them. This is an inference based on the quantity rule

Rule 03:

If

a character wants to control a quantity of a consumable object,

then

he wants to do the plan associated with the object as many times as the quantity he wants will allow.

Since John wanted two apples in (12), the reader can use Rule 03 to infer that John was going to do the plan EAT twice, once with each apple. However, this rule would not be applicable in (13) because a screwdriver is not a consumable object, i.e., the reader would not infer that John wanted to screw in two screws. Since the quantity rule is not applicable to non-consumable functional objects, the understander has to look for other explanations. For example, it might be that the screwdrivers were of different sizes, or that John wanted to keep one in his car and one in his house.

Since acquiring a quantity of a consumable object will fulfill a precondition for a number of goals, possessing a quantity of a consumable constitutes a form of goal subsumption state. Thus Rule 03 above is a way of determining that a character is trying to subsume a number of goals for which the consumable object may be used. In contrast, non-consumable functional objects are usually related to goal subsumption via owership. That is, a non-consumable may be used repeatedly for the same kind of task. Thus acquiring ownership of the object will enable a planner to subsume recurring goals for which the object is suited. This fact about goal subsumption can be expressed by the following rule:

Rule 04:

If

a character wants to own a non-consumable functional object,

then

he wants to subsume those goals for which the object's purpose is suited.

Rule 04 is used in story understanding to infer that someone's action is a plan to subsume a recurring goal, or to infer that a character anticipates having a number of goals for which the functional object may be applicable. For example, recall story (3) above:

(3) John got a job in Glastonbury. He decided to buy a car.

A reader can use Rule 04 to infer that John decided to buy a car in order to subsume the recurring goal of being at work each morning. Goal subsumption was discussed in Chapters 4 and 5.

11.3.1.1 Some More Examples

Some other examples of physical consumption and the memory representations needed to deal with them are:

```
USE-VEHICLE(PLANNER,VEHICLE,LOCATION,FUEL)

  CONSEQUENCES:

    [ PLANNER BE PROX(LOCATION)
      VEHICLE BE PROX(LOCATION)
      FUEL TOWARD NON-EXISTENT        ]

  CONSTRAINTS:

    [ PLANNER IS HUMAN
      PLANNER PROX(VEHICLE)
      VEHICLE IS some vehicle
      VEHICLE CONTAIN(FUEL)
      FUEL IS fuel for VEHICLE        ]
```

The USE-VEHICLE planbox changes the state of the FUEL role filler to non-existent. Since one of the constraints on this plan is that the VEHICLE contain FUEL, using a vehicle will eventually consume its fuel supply so that more fuel will have to be gotten in order to use the plan again.

```
WRITE(PLANNER,SYMBOLS,IMPLEMENT,INK-OBJ,PAPER-OBJ)

  CONSEQUENCES:

    [ INK-OBJ TOWARD NON-EXISTANT
      PAPER-OBJ CONTAIN(SYMBOLS)      ]

  CONSTRAINTS:

    [ PLANNER IS HUMAN
      PLANNER POSS(IMPLEMENT)
      PLANNER POSS(PAPER-OBJ)
      IMPLEMENT CONTAIN(INK-OBJ)      ]
```

WRITE is a plan for communicating by making symbols on a substance by the application of another substance via an implement. Above is shown that WRITE causes the supply of the INK-OBJ to become non-existant, and that having INK-OBJ in the IMPLEMENT is a precondition for the plan.

Note that the PAPER-OBJ may or may not be physically consumed by the plan. In the case where the INK-OBJ is ink and the PAPER-OBJ is paper, the paper cannot be used again and would therefore be consumed. However, in the case of a blackboard, the PAPER-OBJ is retrievable, but only at the expense of losing what was previously written. This situation is discussed below in the section on continuous enablement consumption.

### 11.3.2 Social Consumption

Some consumable objects are consumed in a somewhat less drastic manner than are physically consumable objects. For example consider

(14) John bought a toy from Mary for a quarter. Then John used the quarter to buy a coke.

(15) John bought a toy from Mary for a quarter. Then Mary used the quarter to buy an ice cream cone.

Story (14) is strange in the same way (9) is. It is hard to understand how the objects in these sentences could have been used again. However, (15) is perfectly understandable. The difference between this case and the case of (10) above is that here the object is consumed socially rather than physically. In a plan that socially consumes an object, the object usually changes hands. Since it is no longer controlled by its previous owner, the object is consumed as far as that individual is concerned.

As with physically consumable objects, social consumption is computed from information about the plan in which an object normally participates. For example, MONEY was defined above as a functional object used for the plan BARGAIN-OBJECT. Under BARGAIN-OBJECT is stored the consequences of using that plan on the bargained object:

```
BARGAIN-OBJECT
 CONSEQUENCES:
   [ OBJECT IS POSS(PERSUADEE) ]
```

That is, a result of succeeding to get something from someone by bargaining is that that person now has the possession of the object that was offered. Thus when a plan like BARGAIN-OBJECT is encountered, the object transferred is marked to indicate its current owner. BARGAIN-OBJECT has a precondition that the object bargained must belong to the person bargaining it away. That is, the object must obey the constraint that it be owned by the planner.

This requires that we add to the memory representation the constraints on the slots of the plan. That is, bargaining should contain the following information

BARGAIN-OBJECT

CONSEQUENCES:

[ OBJECT IS POSS(PERSUADEE) ]

CONSTRAINTS:

[ OBJECT POSS(PLANNER)
PLANNER IS HUMAN ]

Now the particular object bargained away cannot be used by its previous owner because it violates the constraint imposed by the plan of bargaining. In sum, the following rule holds for socially consumed substances:

Rule 05:

If
a person uses a socially consumable object in a plan

then
that object cannot be used again by that person for that plan.

In example (14) this rule is violated because the money was consumed as far as John was concerned. (15) is plausible, however, because Rule 05 does not prevent the object from being used again in general, but only from being used again directly by its previous owner.

### 11.3.3 Continuous enablement consumption

There is a third way in which an object can be consumed. Consider the following story

(16) John ran out of thumbtacks putting up posters. He still had an important notice to put up, so he took down one of the posters.

Thumbtacks are consumed in the sense that once one is used, it cannot be directly re-used for the same function without undoing its previous function. However, unlike the other types of consumption we have seen, thumbtacks can be re-used if the planner is willing to undo one of the goals which the tack was used to fulfill. The tack is consumed by its plan in that the tack is needed to continuously enable a goal state (The notion of continuous enablement is discussed in Abelson (1973) and Rieger (1975a)).

This form of consumption requires that the understander compute those states that are continuously enabled by the use of an object in a plan. For example, a tack might have the following functional description:

TACK

FUNCTION:

[ SPEAR(PLANNER, OBJ1, OBJ2, TACK) ]

SPEAR is a plan for connecting two objects by piercing them with a sharp object, in this case, a tack. Associated with this plan would be the following consequences:

SPEAR(PLANNER, OBJ1, OBJ2, PIERCER)

CONSEQUENCES:

```
[ PIERCER BE THRU(OBJ1,OBJ2)
            / \
            ||| cont-enable
            |||
  OBJ1 PHYS-CONT OBJ2             ]
```

That is, a consequence of SPEAR is that the object used to pierce the other two objects continuously enables them to be in physical contact. Attached to the plan SPEAR, then, is a routine that records the state continuously enabled by the particular spearing object. Along with this state would be a pointer to the goal that gave rise to the state. If this object were already continuously enabling a state, this routine would remove this state description from the representation of the object. By doing so, the following rule would be implemented:

Rule 06:

If

a person uses an object that is consumed by continuous enablement,

then

that object cannot be used again without undoing the effects of the plan for which it was last used.

Also, since the representation keeps track of what goal the state was supporting, the understander could infer that the character would have the goal of achieving that state again. This inference requires the rule

Rule 07:

If

1. a state came into existence because it was a goal of a character,

and

2. the state becomes undone

and

3. the character who effected the state learns it has been undone,

then

that character may once again have the goal of bring about that state.

(Note - Rule 07 is actually a general rule not restricted to dealings with objects).

For example, consider the story

(17) John had just used his last thumbtack to put up a poster when his boss told him to put up a notice right away. John took down the poster and put up the notice. Then John went to the store to buy some more thumbtacks.

Q) Why did John want some more thumbtacks?
A) To put the poster back up.

To answer this question, the understander has to infer that John took down the poster to reclaim the thumbtack. Since this undid a state that was once a goal of John's, Rule 07 could be used to infer that John would want to put up the poster again. Having inferred this goal, the last sentence of (17) can be interpreted as part of a plan for this goal, since Rule 06 states that the tack just used is no longer available.

11.3.3.1 Examples

Writing can be a case of continuous enablement consumption. A description of the plan WRITE was given in the section on physical consumption above. In that description, when the PAPER-OBJ is a blackboard, the PAPER-OBJ is consumed until it is erased. After it is erased, the PAPER-OBJ may be used over again, but the function it was previously serving (i.e., displaying the written symbols) stops being served. Thus writing on a blackboard consumes the blackboard by continuous enablement.

Another example of this type of consumption occurs with objects that support or contain other objects (see Lehnert (1978) for a description of support/contain). A bookcase, for instance, keeps books displayed conveniently by providing physical support for them. We would need in memory a description of a bookcase:

```
BOOKCASE
  FUNCTION:
    [ SUPPORT(PLANNER,BOOKS,BOOKCASE) ]
```

That is, a bookcase plays the role of the supporting object in the plan SUPPORT. SUPPORT is a planbox in which the planner places an object into or on top of another in order to support or contain that object. SUPPORT must have associated with it:

```
SUPPORT(PLANNER,SUPPORTEE,SUPPORTER)
  CONSEQUENCES:
   [ SUPPORTER CONTINUOUSLY-ENABLE(SUPPORTEE BE
                                    (TOP SUPPORTER)) ]
  CONSTRAINTS:
    [ PLANNER IS HUMAN
      SUPPORTEE has room
      SUPPORTER IS STRONG      ]
```

This diagram states that the supporting object must have room for the object to be supported. Filling a bookcase with books consumes the bookcase because of the precondition that a supporting object have room for the supportee. To display some other books, the books currently occupying the bookcase must be removed. To understand a story in which a character wishes to use an already filled bookcase to hold some new books, the reader would need knowledge that the bookcase is consumed by continuous enablement to understand why that character would have to un-support the old books to support the new ones, or possibly buy another bookcase.

## 11.4 Conclusions

### 11.4.1 Recapitulation

Processing stories involving objects requires knowledge of the functions normally associated with objects. The <u>functional description</u> of an object specifies the plans that an object is used for, and how it may be used in those plans. This description is useful in determining what plan a character may be using once it is learned that the character wants to possess a particular object.

Each plan contains a description of the consequences of that plan on objects it involves, and the constraints that the plan imposes on the objects it uses. This knowledge is needed to determine if an object is consumed by its use in a plan. There were three distinct ways an object could be consumed: The object could be changed physically; the object could change hands; or the object could be continuously tied up in maintaining a goal state. Consumption is important when stories involve quantities of objects or more than one goal involving an object. Since stories involving goal subsumption (see Chapters 4 and 5) involve multiple occurrences of similar goals, the notion of consumption figures prominently in the analysis of these stories.

### 11.4.2 Objects and Language

The functional descriptions of objects given above were used to explain how objects needed to be thought about for planning purposes. In addition, these descriptions constitutes an important part of the definitions of those words that refer to objects. For example, Nelson (1974) showed that in children the functional aspects of objects constitute the "core" meanings to which identificational features are attached. Miller and Johnson-Laird (1976) present a number of detailed arguments as to the importance of function in representing objects. Despite its seeming primacy in people's memory representation of objects, the functional aspect of the meaning of nouns has been largely ignored by researchers in the semantics of natural languages.

For example, the linguist semantic feature system of Katz and Fodor (1963) attempted to enumerate semantic features of nouns without regard to the function of the objects to which the nouns referred. Katz and Fodor's system has since been demonstrated to be inadequate for its intended purpose (see McCawley, 1968 and Wienreich, 1970, for example). However, the conclusion that Fodor drew from its failure is that it is not possible to break down nouns into semantic features (Fodor, 1975).

Fodor denies, for instance, that it is possible to write down a set of features for the word "chair." His argument is based, essentially, on the fact that the physical nature of a chair can differ radically from one instance of chair to another. So radically, in fact, that no set of features would account for all the possible physical appearances of chairs.

One reason it is hard to write down a physical description of "chair" is that the word "chair" does not strictly refer to a physical object. Rather, part of the meaning of "chair" is its functional description: "A chair is a physical object whose function is for one person to sit down and lean back." This functional description can be used in the mental process that determines whether a given object was an instance of "chair". That is, it might not be possible to specify a set of physical parameters that can distinguish all objects that are chairs from those that are not. But a functional description of a chair could still be used by an understander to compute whether or not a particular object should be called a chair.

For example, in a fairy tale, a reader might encounter a magic chair that had no legs but which could be used to sit on nevertheless. Fodor would point out, correctly, that any set of physical characteristics we had previously assigned to the word "chair" would almost certainly be violated by this object. However, the functional description of "chair" would not be violated. It was still an object one could use to sit on and lean back, and hence the use of the word "chair" can be computed to be applicable.

The lack of success of Fodor's research program is not an argument against the plausibility of finding representations for the meanings underlying words. Rather, it is evidence that a theory of meaning must take into account the "human" as well as the physical nature of the world. Words often refer to events or objects in terms of the role that these things play in human affairs. Nouns of functionality are an example of words whose meaning can only be described in terms of the intentional nature of the human world.

# CHAPTER 12

# DIFFICULTY

Stories are often about fulfilling troublesome goals. To understand these stories, a definition of difficulty is required. This chapter provides two definitions, one for the expected difficulty of a plan, and another for the experiential difficulty of executing a plan. A plan is expected to be difficult if it is resource-intensive, if it is likely to generate difficult preservation goals, or if part of the plan is expected to be difficult. A character will experience difficulty executing a plan if the plan is expected to be difficult, if it requires more resources than were anticipated, or if the planner tries to execute a plan without having first fulfilled a precondition for the plan.

These definitions of difficulty are useful for understanding a number of story situations. In particular, they are used to infer that a character who is experiencing difficulty fulfilling a recurring goal may want to establish a subsumption state for that goal.

## 12.1 Introduction

The notion of difficulty has arisen several times in the preceeding chapters. As was mentioned in Chapter 3, a difficulty in fulfilling a goal poses a problem for a story character. Thus difficulty gives rise to interesting story situations. In addition, difficulty plays a role in other types of stories. In the chapter on goal subsumption, I showed that repeated difficulty in fulfilling a recurring goal is a reason for wanting to subsume that goal. For example, consider the story:

(1) It was taking John three hours to ride his bicycle to work in the morning. He decided to buy a car.

John is having difficulty getting to work in the morning because riding a bicycle for three hours is a difficult plan for this goal. A reader uses the fact that John is having trouble fulfilling this goal

to explain the next sentence as an instance of goal subsumption replacement (see Chapter 5). Having difficulty fulfilling a recurring goal usually predicts that the planner will try to establish a new subsumption state for that goal to eliminate the difficulty. Since owning a car subsumes the goal of getting to work, and replaces the function of the bicycle, the reader infers that John bought the car to subsume this goal.

In contrast, suppose the story were the following:

(2) It was taking John five minutes to ride his bicycle to work each morning. John decided to buy a car.

Since no difficulty is implied by the first sentence, it is unlikely that John is trying to re-subsume a goal in the second. Another explanation for John's behavior is needed. The reader might infer that John bought the car to take trips, or to pick up groceries, but not to commute to work.

Thus to understand the reason behind John's behavior in story (1), a reader must recognize that John is experiencing a difficulty. However, there is no explicit mention of difficulty in the story. The reader must infer the difficulty from the description of John's activity. To make this inference, a reader needs a notion of what it means to execute a difficult plan, and must be able to detect difficulties when they appear in a story text.

This chapter suggests a definition of difficulty, and describes an algorithm by which difficulty can be computed. This definition is useful for identifying difficulties as they arise in story situations, and defines a class of interesting story situations based on the expected difficulty of a character's goal.

## 12.2 The Notion of Difficulty

There are two aspects of difficulty that are needed for story understanding. First, there is the notion of experiencing difficulty. This term is descriptive of the actual execution of a plan. The second aspect of difficulty refers to expected difficulty. Expected difficulty is used to describe the anticipated problems that one will experience trying to execute a plan. For example, consider these stories:

(3) John had to wait for an hour on the checkout line in order to buy his groceries.

(4) John wanted to buy some groceries.

(5) John wanted to climb Mount Everest.

Story (3) describes a situation in which a character has difficulty buying groceries. That is, the actual execution of this plan turned out to be problematic. However, the plan itself is not expected to be difficult. For example, consider story (4). Mentioning this goal does not cause most readers to assume that John will have trouble trying to carry out his plan. Thus story (3) contains an instance of experiencing difficulty that was unexpected.

In contrast, story (5) does contain an expected difficulty. While the reader has not yet heard of John's travails, he expects John's task to be hard under any circumstances.

To spot either kind of difficulty in a story, a reader needs a definition of difficulty. The definition for expected difficulty is given first, since this definition is used within the definition for experiencing difficulty.

## 12.3 Expected Difficulty

The following stories all refer tasks that are expected to be difficult:

(6) John wanted to become dictator of the United States.

(7) John wanted to become invisible.

(8) John wanted to get a certain postage stamp to complete his collection. Then he learned the stamp cost one million dollars.

(9) John decided to try to play tiddlywinks in the middle of a busy intersection.

Becoming dictator of the United States is hard because it requires a great deal of power and is likely to be met with opposition. Becoming invisible is difficult because there are no known plans for it. John's task is difficult in story (8) because of the high cost of the desired item. Playing tiddlywinks in a busy intersection is problematic because it is hard to do without interference.

To find the expected difficulties in these stories, two definitions are needed, one for difficult goals, and the other for difficult plans. For example, wanting to be invisible is expected to be a difficult goal since there is no known way to bring it about. Buying a stamp for a million dollars is a difficult plan, because it requires its user to have a great deal of money. I propose the following definitions for difficult goals and plans:

Expected Difficulty

A goal is expected to be difficult if its planner has no plans for it, or if all its plans are expected to be difficult.

A plan is expected to be difficult if either:

1. The plan requires more resources (capabilities) than the understander believes is typical for a plan for that goal.

2. The plan requires more resources (capabilities) than appear to be available.

3. The plan is expected to generate difficult preservation goals.

4. A part of the plan is expected to be difficult.

Each of these categories is now treated in detail.

### 12.3.1 More than typical resource requirements

A plan can be difficult is it requires more resources than the understander believes to be typical for a plan for that goal. This implies that the understander has in its memory specific information about the amount of resources required by a plan for a particular goal. For example, consider the story (1) above:

(1) It was taking John three hours to ride his bicycle to work in the morning, so he decided to buy a car.

Most readers of story (1) will assume John is spending too much time getting to work, i.e., that his particular plan for his "be at work" goal is requiring too much time. Making this inference requires that the understander have a concept of the amount of time people should allot to getting to work. It is then a straightforward matter to determine that the time expenditure in (1) is beyond the acceptable limits.

Thus the reader must possess knowledge about the typical amount of resources that are expended fulfilling a goal. For example, under the goal "be at work" should be stored the information that this goal should normally take at most an hour or so to achieve. When the reader learns how much time it is actually taking John to fulfill this goal, the reader can determine that this is greater than normal for that goal.

The notion of belief is important here because I want to include as difficult those things that everyone agrees require too much of a resource. For example, finding a parking space in Manhattan is termed difficult because people have a preconceived notion of how much time it is worth spending to find a parking space, although this ideal may never be realized.

Also, it is important that the difficulty of a plan is defined relative to its goal. For example, if John were bicycling to work and taking three hours, this would be considered a difficult plan. However, if John's goal was to enjoy recreation, then the plan would not be found difficult by this rule.

### 12.3.2 More than available resource requirements

Another way in which a plan may pose a difficulty for a character is if it requires more resources than appear to be available, i.e., if the plan is beyond the capabilities of the planner. For example, consider the following stories.

(10) John wanted to buy a mansion, but he was earning an assistant professor's salary.

(11) John wanted to buy a house, but he was on welfare.

(12) John realized that he had to jump over the 10 foot fence to get away from the man who was chasing him.

(13) Spassky knew he was in for trouble when Fischer began to reveal his brilliant strategy.

In (10), the limitations of financial resources would make the payments on a house questionable. This is true in (11) as well because the cost in relation to resources is great, in spite of the fact that the absolute costs and incomes in the two examples are different.

In (12), the extent of John's physical prowess appears to be in question. Since the task involved is generally thought to be at the extreme end of a person's physical abilities, the plan of jumping over the fence would be characterized as difficult.

Sentence (13) has a difficulty on the level of mental rather than physical resources. Winning a chess game requires the mental faculty to outwit one's opponent. Since Spassky appears to have a lesser faculty than Fischer, his winning the game is a difficult task.

To compute the difficulty in each of these situations, the reader must have stored in memory the amount of resources typically consumed by a plan. For example, the memory representation for "mansion" must show that buying a mansion normally consumes an enormous amount of money. The representation for an ordinary house must show that it consumes a large amount of money.

Similarly, the amount of resources possessed by types of people must be recorded. For example, an assistant professor might be shown to have a greater than average intelligence and a moderate income; a person on welfare would be shown to have a very small income. If the resources needed are much greater than those available, then a difficulty is present.

This computation is applicable to abilities as well as resources. For example, to compute the difficulty in story (12), a reader needs to know the normal physical abilities of a person, and compare them to the task at hand. Suppose the memory representation for a person showed that people can normally jump no more than three feet off the ground. Since this is much less than is required for John's task, his plan is a difficult one.

### 12.3.3 Generating difficult preservation goals

A plan for a goal may be difficult if the execution of that plan is likely to generate difficult preservation goals for a character. These situations were discussed in Chapter 7 on goal conflict resolutions, and in Chapter 9 on goal competition resolutions. For example, consider the following stories:

(14) John wanted to beat up Muhammad Ali.

(15) John wanted to wade across the swamp, but it was full of hungry mosquitos.

In story (14), John's task is formidable because it is unlikely that he could execute his plan without angering Ali. Since an angry Ali is capable of hurting John, John would then have the goal of preserving his health. Against Ali, this goal is expected to be difficult due to the likely mis-match of fighting abilities. Thus John's original plan is expected to be difficult because it is likely to generate a difficult preservation goal.

In story (15), John's plan for crossing the swamp has the consequence of placing John in the proximity of the hungry mosquitos, which is likely to be rather irritating. Being bitten by mosquitos is undesirable, and very difficult to prevent if one is surrounded by lots of mosquitos. Thus John's plan will generate a difficult preservation goal upon execution, and so wading across the swamp is expected to be a difficult plan.

Detecting this form of difficulty involves determining that a preservation goal will be generated, and then judging the difficulty of that goal. Detecting a preservation goal was discussed in chapters 7 and 9. The expected difficulty of this goal can be computed by the same techniques for determining the difficulty of other types of goals.

### 12.3.4 Difficult plan components

A plan is difficult if it has a precondition which is either difficult to achieve or difficult to maintain. A difficult precondition means that achieving that precondition is expected to be difficult. For example, consider

(16) To rescue Mary, John had to get past a ferocious dragon.

(17) It's hard to change a flat on the turnpike because there's no place to pull over.

Since getting past a ferocious dragon is likely to be beyond the limit of John's means, getting past the dragon is a difficult precondition for John's goal of saving Mary. By the above rule, rescuing Mary is difficult. In (17), changing a flat is difficult because of a resource limitation on the precondition of being in a safe place. Likewise, in

(18) John had trouble sleeping because he lived near the airport.

John most likely has no plan to effect the tranquility that is normally a precondition for sleeping. Hence, for John, Satisfy-sleep by sleeping is a difficult plan.

Contrast (17) with

(19) John had trouble sleeping because whenever he dozed off, his wife began to snore.

John probably has a plan for the instrumental goal of restoring tranquility in this situation (i.e., nudge wife). The problem with this plan is that the state it achieves (wife be quiet) is temporary - it is likely to spontaneously degenerate over time. I call such a state an unstable precondition. If a plan requires as a precondition that a state be continuous, then an unstable precondition makes for a difficult plan. This is the case in (18), where tranquility must be continuous in order to sleep, and where the wife not snoring is an unstable state. Thus for John, getting some sleep is difficult because of the unstable precondition of maintaining tranquility.

In most cases of this type of difficulty, the presence of a difficulty is made explicit in the text. For example, in story (18), the reader is told that John is having trouble sleeping. To determine the actual cause of difficulty, the reader must examine the reason given in the text. The reason given in story (18) is living near the airport. One of the consequences of living near the airport is that it is noisy. The reader must check to see that this state violates a precondition for John's plan. Since being noisy precludes the tranquility required for sleeping, the reader then checks to see if this is a difficult precondition. Since John has no control over the airport noise, there is nothing he can do about this problem. Thus John's task is likley to be a difficult one.

In sum, a goal is expected to be difficult if it has no plans, or if all its plans are expected to be difficult. Difficulty plans either require more resources than seems reasonable or available, or they generate preservation goals during their execution, of they have subparts that are expected to be difficult.

## 12.4 Experiencing Difficulty

There are a number of ways in which a character can experience difficulty executing a plan. For example, consider the following stories:

(20) John tried jumping over a ten foot tall fence.

(21) John went into the supermarket to buy a candy bar, but he had to wait for forty-five minutes on the checkout line.

(22) John's mattress was so lumpy it kept him awake nights. He decided to buy a waterbed.

John experiences difficulty in all these stories. In story (20), John had trouble executing a hard task; in story (21), he met with an unexpected delay; and in story (22), the quality of the mattress prevented John's plan from working.

To recognize the difficult present in these examples, the following definition is offered:

Experiential Difficulty

A character has experienced difficulty executing a plan when

1. The plan is a difficult plan.

2. The plan may take more of a resource than was anticipated.

3. The character tried to execute the plan without having first fulfilled a precondition of the plan.

For example, John probably ran into trouble in story (20) because he was trying to execute a plan that is expected to be difficult. In story (21), the supermarket could not process John through fast enough, so his plan to get a candy bar was more difficult than he expected. In story (22), one of the preconditions required for sleeping is to be on a comfortable surface. Since John tried to execute the plan without first fulfilling this precondition, he had a hard time of it.

Applying these criteria is done as follows: The first criterion is just a matter of checking for a difficult plan underlying the character's actions. This can be done by inferring the plan, and then using the knowledge described in the last section to determine if the plan is expected to be difficult.

The reader checks for difficulty based on using more of a resource than was anticipated if the event specifies the amount of the resource actually used. For example, in story (21) the reader is told that John took three quarters of an hour to execute his plan, which is longer than the normal amount of time required for this plan. In effect, this

criterion is tested by using the procedures for expected difficulty on the actual data supplied by the story.

If the fact that a character is having a problem is explicit in a story, then the reader can also check to see if the story mentions an explicit precondition that has not been met. For example, in story (22), John's mattress fails to meet the precondition of the plan for sleeping because it is not comfortable, and thus the reader can compute that this is the reason for his difficulty sleeping.

## 12.5 Application

Having defined the notion of difficulty, let us apply it to goal subsumption situations involving repeated difficulty fulfilling a recurring goal. In Chapter 5, the following rule was given for these situations:

Rule GS5:

If

a character has repeated difficulty fulfilling a recurring goal,

then

that character may have the goal of establishing a subsumption state that eliminates the difficulty.

*in order to apply GS5, an understander needs to know that a* difficulty, as I defined above, exists in the story. This can be accomplished by the following algorithm:

1. Determine if a difficulty is explicitly mentioned. For example, in consider story (22) above:

   (22) John's mattress was so lumpy it kept him awake nights. He decided to buy a waterbed.

   The first sentence of this story can be represented as

```
mattress  BE PHYS-STATE(-)
   |       / \
 John's    |||  leadto
           |||
        John BE CONSCIOUSNESS(0)
          MODE - CANNOT(5)
```

Here I allow modes to take values from 0-10 in order to represent varying degrees of disability. Thus the above representation states that the mattress being in a poor physical condition disabled John from sleeping. In this case, it is necessary to check only that the caused conceptualization is an event dominated by a goal to

determine that a difficulty exists for the character. That is, something is disabling John from sleeping, and sleeping is usually dominated by Satisfy-sleep. Since Satisfy-sleep is a recurring goal, rule GS5 states that John may have the goal of subsuming Satisfy-sleep by eliminating the difficulty.

2. The difficulty may be an inference about a character's belief or knowledge state. For example, consider:

(23) When John saw Bill coming over with his knife in his hand, he knew he was in trouble.

(24) John was stunned when he heard how much a new car would cost him.

In these stories, a character expresses an explicit belief about the difficulty of an event. If the understander believes the character (i. e., he does not believe that the character has bad judgment or has little confidence in himself), then the understander can infer the difficulty the character believes. In story (23), John believes he will have difficulty protecting himself from Bill, and since the understander has no reason to doubt John, it can infer that John will have a difficulty. In story (24), John's reaction to the cost of a car indicates that John thought he would have some trouble affording one. Thus the reader can infer that buying a car presents John with a difficulty.

3. The difficulty may have to be inferred from a statement about the amount of resources required for a plan (i.e., it a statement about the amount of time required, the amount of money needed, etc.). This is the case in story (25):

(25) The property taxes in John's neighborhood began to exceed his income. John decided to move.

Here the amount of money required to met the tax payments makes living in his current home difficult for John. To find the difficulty here, the reader must apply the difficulty heuristics described above. For example, one of the heuristics given above states that a character will experience a difficulty if he tries to perform a plan that is expected to be difficult. One reason why a plan might be expected to be difficult is if it requires more resources than are available. Since this is likely to be the case if John must pay out more money than he earns, the reader can infer that maintaining his house requires John to execute a difficult plan. Thus John must be experiencing difficulty in story (25).

## 12.6 Summary

The notion of difficulty arises in a number of story situations. To determine that a character has a difficulty, it was necessary to distinguish between two types of difficulty. First, there is expected difficulty, which is used to describe goals and plans whose fulfillment and execution, respectively, seem likley to cause a planner a problem. Second, there is experiential difficulty, which refers to the trouble a person actually experienced trying ot execute a plan.

A goal is expected to be difficult if its planner has no plans for it, or if all its plans are expected to be difficult. A plan is expected to be difficult if the plan requires more resources or capabilities than the planner has available, if it requires more than the understander believes is typical for a plan for that goal, if the plan is expected to generate difficult preservation goals, or if part of the plan is expected to be difficult.

The experience of difficulty could come about in a number of ways: The plan being executed may be a difficult plan; the plan be using up more of a resource than was anticipated; or the planner may be trying to execute the plan without having first fulfilled one of its preconditions.

These definitions of difficulty are useful for story understanding. For example, a story understander often needs to infer that a character is having difficulty fulfilling a recurring goal in order to infer that he is trying to subsume that goal. The inference that the character is experiencing difficulty can be made using these definitions, thereby enabling the understander to infer the character's desire to subsume a goal.

## CHAPTER 13

## HOW PAM WORKS

### 13.1 Introduction

The previous chapters of this thesis have been concerned with the knowledge needed for story understanding and with the algorithms that use this knowledge. This chapter describes the actual PAM implementation of these processes.

PAM is primarily concerned with using its knowledge to find explanations. As was described previously, PAM's explanations consist of intentional entities: A plan explains an event; a goal explains a plan; and a goal is explained by a theme, by instrumentality, or by the need to subsume a recurring goal. In the course of finding these explanations, PAM may be required to spot a conflict among a character's goals, to detect competition among the goals of different characters, or to determine that several characters' goals are in concordance.

PAM's explanation algorithm, described in Chapter 2, makes use of two components: A bottom-up mechanism that can make explanatory inferences from an input independent of context, and a predictive mechanism that constrains the bottom-up inference process. The predictive mechanism tries to find an intentional explanation by relating an input to the story representation. If it cannot, the bottom-up mechanism is used to suggest possible explanations for the input. The predictive mechanism then tries to find an explanation in the story representation for the inferred explanation. The cycle repeats until either a connection to the story representation is established, in which case an explanation has been found, or until no more bottom-up explanatory inferences can be drawn. In this case, the program fails to understand the input. This process is outlined in Figure 12.

There are two major problems in implementing this algorithm. First, the amount of knowledge required for the task is large. The bottom-up mechanism must be capable of efficiently accessing the knowledge relevant to the particular sentence at hand. Second, once

Figure 12
PAM's Processing Algorithm

```
              Sentence in Conceptual Dependency

                         ||
 //===============>      ||
 ||                      \/
 ||
 ||    Predictive component:
 ||
 ||     __________________________
 ||    |                          |   yes
 ||    | Is there a prediction    |  ====>  algorithm
 ||    | that explains the input? |         terminates
 ||    |__________________________|         successfully
 ||
 ||                      ||
 ||                      || no
 ||                      \/
 ||
 ||    Bottom-up component:
 ||     __________________________
 ||    |                          |   no
 ||    | Can an explanation be    |  ====>  algorithm
 ||    | inferred from the input? |         terminates
 ||    |__________________________|         unsuccessfully
 ||
 ||                      ||
 ||                      ||
 ||                      \/
 ||
 ||          Inferred Explanations
 ||
 ||                      ||
 \\=================//
```

relevant knowledge has been accessed, a means is needed to use that knowledge to make predictions about subsequent sentences. That is, the program needs to know what predictions to make, to determine whether any have been met, and if so, to decide what should be put in the story representation.

### 13.2 Overview

As the earlier chapters of this thesis indicated, the knowledge needed for story understanding can be usefully expressed as rules, or condition-action pairs. Most of the knowledge in PAM is in fact kept in the form of rules. (For an overview of rule-based systems, see Waterman and Hayes-Roth (1978)). PAM's processing can be viewed largely in terms of finding the rules relevant to a particular input, and performing the functions that those rules state.

PAM uses rules to build up a story representation. To explain how PAM does this, this section provides a highly simplistic description of PAM's processing. A very simple example is used to give a general overview of PAM's mode of operation. The subsequent sections of this chapter will describe each aspect of the processing in greater detail.

Consider the following simple story:

(1) John was hungry. He ate at a restaurant.

The representation PAM produces for this example should include the following elements: The story has a theme, John's being hungry, which generates John's goal of satisfying his hunger. To achieve this goal, John chose the plan of doing the restaurant script. Then John executed this plan by eating at a restaurant.

Thus a representation for this story might look like the following:

```
Theme - Hunger
  |
  |
  |
Goal  - Satisfy-hunger
  |
  |
  |
Plan  - Do-restaurant-script  ----> Action
                                    (John eat)
```

To build this representation, PAM has to make the inference that John has the goal of satisfying his hunger, and that eating at a restaurant is a plan for this goal. These inferences require the following knowledge:

Rule 1:

If

a person is hungry

then

that person may have the goal of satisfying his hunger.

Rule 2:

If

a person eats something,

then

that action may be a plan to satisfy his hunger.

Using these rules to build the story representation imposes the several requirements: First, the program must determine that these particular rules are applicable to this story. As the previous chapters have indicated, PAM needs a large number of rules for story understanding. Since the program cannot know in advance which rules will apply to the sentences of a given story, PAM must have some way of selecting these rules upon reading the story text.

Second, applying these rules requires that they be used to build a representation. For example, Rule 1 above implies that a character has a goal, but in an actual program, the rule must be used to build a piece of story representation denoting this goal. The major difficulty here is in getting all the rules to contribute to the same story structure. For example, Rule 1 above is responsible for adding a goal to the story representation, and Rule 2 for adding a plan. The problem is to link the structures built by each rule together into the representation shown above.

The problem arises because the mechanism being described is used to implement contextual understanding, and thus must be built out of components that are context free. That is, each rule is designed to be independent of the other rules of the system. For example, Rule 2 represents an explanation for the second sentence of story (1) that is independent of whether or not PAM had previously learned of John's goal; if it had not, then PAM would have to infer this goal. In any case, a method is needed to tie the structures generated by different rules into the same story representation.

With this problem in mind, consider how PAM processes the above story. PAM reads in the first sentence, which has the following representation in Conceptual Dependency:

((ACTOR HUM0 IS (*HUNGER* VAL (-3))))

According to the algorithm in Figure 12, when PAM encounters an input sentence, it first checks to see if any predictions exist that are confirmed by the sentence. Since PAM is processing the first sentence of a story, it cannot have a prediction readily available with which to interpret the input. Hence PAM must enter the bottom-up phase, and determine if it has any rules applicable to the input. A rule is applicable to an input if the condition of the rule is satisfied by that input. Thus PAM examines its rules to find those whose condition is satisfied by this input. Let us ignore for the time being how this process can be performed efficiently, and return to this question later on.

PAM finds that Rule 1 has a condition that is met by the input. Rules in PAM are stored as LISP atoms with conditions, etc., kept as properties of those atoms. For example, the PAM form for Rule 1 might be called HUNGER-RULE, and has the following condition:

HUNGER-RULE

```
Condition:   [LAMBDA (X)
                (AND (E (PATH '(IS) X) '*HUNGER*)
                     (LESSR (PATH '(VAL) X) 0) ]
```

Each condition predicate is a function of one argument. This predicate makes use of some simple PAM functions: PATH is a function for finding the fillers of the roles of a conceptualization. That is, both Conceptual Dependency structures and the structures PAM generates consist of a set of roles and associated role fillers. The input conceptualization has two roles, an ACTOR role and an IS role. The filler of the ACTOR role is HUMO, and the filler of the IS role is (*HUNGER* VAL (-3)). This latter structure has the role VAL, whose filler is (-3). PATH is used to access the fillers of roles by specifying a list of roles and a conceptualization. The following are examples of calls to PATH using the conceptualization for the first sentence:

```
(PATH '(ACTOR) '((ACTOR HUMO IS (*HUNGER* VAL (-3)))) )

    returns the value HUMO

(PATH '(IS) '((ACTOR HUMO IS (*HUNGER* VAL (-3)))) )

    returns the value (*HUNGER* VAL (-3))

(PATH '(IS VAL) '((ACTOR HUMO IS (*HUNGER* VAL (-3)))) )

    returns the value (-3)
```

E is a function that tests if the first element of a conceptualization of has a particular value, and LESSR performs a numeric comparison. Thus the condition predicate of HUNGER-RULE shown above is true if the IS role contains the form *HUNGER*, and if the VAL role contains a negative number.

Applying this predicate to the input conceptualization returns true, so PAM now proceeds to evaluate the action of this rule. Rule 1 states that the character has the goal of satisfying his hunger. Thus the action of HUNGER-RULE is written as follows:

```
HUNGER-RULE

  Action:
  [BUILD '((GOAL (*S-HUNGER* PLANNER (NIL))
            SOURCE (*THEME* VAL (*HUNGER-DRIVE*))
            PLAN (NIL) )) ]
```

This action, as all the actions of PAM's rules, builds a structure. The function BUILD produces a structure in PAM's internal format, the

details of which are not crucial to the current discussion. The structure produced by HUNGER-RULE is called a goal episode, and contains three roles: a goal, a goal source, and a plan. The GOAL role is filled with a representation denoting the goal Satisfy-hunger, the SOURCE with the theme of having the hunger drive, and the PLAN role is left empty.

An empty role in a representation is called a gap (Riesbeck, 1975). A gap is a piece of representation that could be filled by subsequent sentences. The gap for a plan role in the structure built by HUNGER-RULE means that goals usually give rise to plans, and that although the plan to be used for this goal is not known, it may become apparent later on in the story.

Thus a gap is a place in a story representation that the understander knows must exist because of the nature of the knowledge it is applying. That is, PAM knows that goals have sources that generate them, and plans that effect them. This knowledge is expressed by structures that contain places for these items even if the particular items themselves are not known.

In addition to building structures that contain gaps, PAM also makes predictions about how those gaps may be filled. For example, a prediction about the gap for the plan in the structure built by HUNGER-RULE is that it will be filled by a structure denoting a plan for satisfying hunger. To use this prediction to fill the gap, it is necessary to state the prediction in the form of a rule: If a plan for satisfying hunger is encountered, then this plan is a plan for John's Satisfy-hunger goal.

Thus we want to set up a prediction that is looking at the input for an appropriate plan, and if such a plan is spotted, place the plan into the PLAN gap already present in the story representation. This is accomplished as follows: Stored along with every rule in PAM is a list of other rules. These rules are used to make predictions about how to fill a gap in the representation created by the original rule. Since these rules suggest how to fill gaps, they are termed suggestions. The idea of attaching suggestions to rules is due to Riesbeck (1975), who used this idea in his conceptual analysis program. He called rules that contained suggestions requests, and I use this terminology here.

Each suggestion specifies the gap in the story representation it is intended to fill, and where it will look to find the filler. For example, the HUNGER-RULE request has the following suggestion:

```
HUNGER-RULE

 Suggestions:   [( (PLAN) !INPUT! SUITABLE-PLAN-RULE)]
```

This suggestion can be interpreted as follows: The first field specifies the role of the structure built by HUNGER-RULE that the suggestion is trying to fill, in this case, the PLAN gap. This field is called the target. The second field denotes the place where the request is to look to find the plan. This is termed the focus. This suggestion specifies that the focus is the special gap !INPUT!, which is the place

in PAM where new inputs get placed. That is, PAM has a gap referred to by !INPUT! that is used to hold new conceptualizations as they are read in or inferred by PAM. Having such a gap allows PAM to make predictions simply by focusing a request on the input gap.

Lastly, the name of the request to be used is given. In this example, the request is called SUITABLE-PLAN-RULE. In sum, this suggestion states that the request SUITABLE-PLAN-RULE should be used to try to fill the target PLAN gap in the structure that HUNGER-RULE builds, and it should do so by examining its focus, the special gap !INPUT! where new conceptualizations are placed.

SUITABLE-PLAN-RULE is a request whose condition will be true if its focus contains a structure denoting a plan for a particular goal. Its action is simply to return this plan structure. Thus SUITABLE-PLAN-RULE has the following form:

```
SUITABLE-PLAN-RULE

 Condition:   SUITABLE-PLAN-PREDICATE

 Action:      !FOCUS!
```

That is, the condition of this request is a predicate that is true if the focus of the request is a plan for the goal in the story representation. The action is represented by the special symbol !FOCUS!. This symbol means that when executed, the action will return the contents of the request's focus.

Recall that the purpose behind making a suggestion is to make a prediction that will help fill a story gap. That is, when a new input is encountered, we want PAM to check the condition of this request to see if the prediction is confirmed, and if it is, to fill in the target story gap. Thus PAM must remember that this request is focused on !INPUT!, and that its target is the PLAN gap in the structure built by the request HUNGER-RULE.

To record this information, PAM makes a copy of the request specified in the suggestion, and adds to it properties specifying the focus and the target. PAM also goes through the condition and action fields of the request, and replaces the special symbols with their actual referents. The copy of SUITABLE-PLAN-RULE thus looks like this:

```
R5

 Condition:   SUITABLE-PLAN-PREDICATE

 Action:      INPUT

 Focus:       INPUT

 Target:      CON12
```

R5 is the copy of SUITABLE-PLAN-RULE. Its focus is INPUT, which is the name of the actual LISP atom where new inputs are placed. That is, the suggestion specified to focus the request on !INPUT!; INPUT is the actual value of !INPUT!, which is used to refer to this slot symbolically (This extra level of indirection has to with the details of PAM's implementation and has no other significance).

The target of R5 is CON12. CON12 is the name of the PLAN gap in the structure built by HUNGER-RULE. Recall that the action of HUNGER-RULE used the function BUILD to create its structure. BUILD creates atoms for every role filler in the structure it generates so that they may be referred to conveniently. For example, the action of HUNGER-RULE is the following:

```
[BUILD '((GOAL (*S-HUNGER* PLANNER (NIL))
          SOURCE (*THEME* VAL (*HUNGER-DRIVE*))
          PLAN (NIL) )) ]
```

The actual structure it returns is some generated LISP atom, say CON9, whose value is as follows:

```
CON9 <-> ((GOAL CON10 SOURCE CON11 PLAN CON12))
```

Each atom in this structure has these values:

```
CON10 <-> (*S-HUNGER* PLANNER CON13)

CON11 <-> (*THEME* VAL CON14)

CON12 <-> (NIL)
```

These atoms, called <u>nodes</u>, are built into the story structure so that pieces of the structure can be referred to and changed simply by referring to an atom or changing its value, respectively. For example, CON12 denotes the gap for a plan in the structure built by HUNGER-RULE. This node is given the value (NIL), which indicates that it is a gap. Other requests such as SUITABLE-PLAN-RULE can refer directly to this node. Request R5 uses this node as its target, so when this request's condition is found to be true, PAM need only set the value of CON12 to the value of the action built by R5 in order to change the story representation.

When a suggestion causes PAM to copy over a request, it replaces all references to the story representation with the actual nodes denoting those parts of the representation. For example, the suggestion made by HUNGER-RULE specifies that SUITABLE-PLAN-RULE should be focused on !INPUT!, and the focus of R5 is made into INPUT, the node to which !INPUT! points. Similarly, the action of SUITABLE-PLAN-RULE is !FOCUS!, and when PAM creates R5, the action is made into INPUT, the node at which the request is focused. Thus SUITABLE-PLAN-RULE and its copy R5 look as follows:

SUITABLE-PLAN-RULE

Condition: SUITABLE-PLAN-PREDICATE

Action: !FOCUS!

R5

Condition: SUITABLE-PLAN-PREDICATE

Action: INPUT

Focus: INPUT

Target: CON12

R5 is attached to the story representation as is shown below:

---

```
((GOAL CON10 SOURCE CON11 PLAN CON12))
                            /\
                     target ||
                            ||

                            R5

                            |
                      focus |
                           / \

                          INPUT
```

---

Thus a suggestion builds a copy of a request, and attaches that request to the story representation by giving it a target gap to fill, and a focus gap to look at. Also, symbolic references in the action and condition of the request are replaced by their actual referents in the story representation. An instance of a request that is attached to a story representation in this manner is called an <u>active request</u>.

By separating out the focus and target fields of a request from the condition and action fields, requests can be stored in memory without specifying where the structures they build should go, or where they should look to see if their conditions have been met. Thus several active instances of the same request may exist simultaneously, each focused and targeted on different locations.

Active requests serve to implement predictions about what will fill a gap, and where the filler might come from. PAM checks to see if a prediction has been confirmed whenever a gap that a request is focused upon is changed. For example, R5 is focused on INPUT, the special gap into which PAM places new inputs. When INPUT is filled, in this case by reading in a new sentence of the story, PAM will check the requests focused on INPUT to see if any of them have conditions that are made true by the input. If INPUT is filled by a plan for satisfying hunger, the condition of R5 will be met, and the action of R5 taken. This action will fill in CON12, the PLAN gap in the story representation, with the plan found in INPUT, thus joining this plan to the previous representation. The following story representation is thus formed:

```
((GOAL (*S-HUNGER* PLANNER (HUM0))
  SOURCE (*THEME* VAL (*HUNGER-DRIVE*))
  PLAN (NIL) ))
```

In order to keep track of which requests are focused on a gap, PAM keeps a list as a property of each node that has requests focused upon it. In the example above, INPUT would have the following property:

```
INPUT

Focused-requests: ( ... R5 ... )
```

That is, among other requests that make predictions about the input is R5. When INPUT is filled, PAM need only check the requests on this list. If PAM finds one whose condition is met, then that request predicted the input. Otherwise there is no prediction that can yet interpret the input.

So far in processing story (1), PAM has encountered the first sentence and found no predictions to interpret it. Then PAM searched for a rule whose condition is met by the input, and found the request HUNGER-RULE. This rule says to build a structure containing a representation for John's Satisfy-hunger goal, and for the theme that generated this goal. The structure also contains a gap for a plan for this goal, which is currently unknown. HUNGER-RULE suggested that this gap could be filled from future inputs, and activated a copy of SUITABLE-PLAN request to make this prediction.

However, PAM has not yet stopped processing this sentence. PAM only stops successfully when it finds some prediction that is confirmed by the input or by an inference drawn from the input. To determine whether this has occurred, PAM places the structure just built by HUNGER-RULE into INPUT, the input gap, and again checks the requests focused on this gap.

Since PAM stops processing only when a prediction is confirmed, it is necessary to have some very general predictions around at the start of story understanding. One such prediction is implemented by an active request whose condition looks for a structure that has a theme in it. This request has the following form:

R1

```
Condition:   (LAMBDA (X)
               (E (PATH '(SOURCE) X) '*THEME))

Action:      INPUT

Focus:       INPUT

Target:      CON7
```

That is, R1 is an active request focused on INPUT, the special input gap. It tests for a structure having a thematic source, and if it finds one, simply moves the structure into CON7, a gap created by PAM at the beginning of understanding for the purpose of containing a whole story episode. This prediction states in effect that PAM can stop processing a sentence when it has built a structure containing a theme.

Since R1 is focused on INPUT, the "Focused-requests" property of INPUT contains R1 in its list. Thus when PAM places the structure built by HUNGER-RULE into INPUT and checks the requests in this list, PAM will find that the condition of R1 has been met. A prediction has now been confirmed, and so PAM can go on to processing the next sentence.

With respect to the processing of the request HUNGER-RULE, there are a number of points which the preceding discussion ignored. For example, the action of HUNGER-RULE was given as:

```
[BUILD '((GOAL (*S-HUNGER* PLANNER (NIL))
         SOURCE (*THEME* VAL (*HUNGER-DRIVE*))
         PLAN (NIL) )) ]
```

Note that the PLANNER role of the Satisfy-hunger goal is empty in the structure built by this action, but it should be filled with HUM0, denoting the character John. PAM uses suggestions for this purpose as well. That is, there is a second suggestion made by HUNGER-RULE:

```
( (GOAL PLANNER) (ACTOR) FOCUS-REQ)
```

The target of the suggestion is the PLANNER role of the ACTOR role in the structure HUNGER-RULE builds. The focus is the ACTOR role of the input conceptualization. FOCUS-REQ is a request that simply moves its focus into its target if the focus is not empty. Thus this suggestion activates a request that will move HUM0 from the ACTOR role filler from the input conceptualization into the gap for the PLANNER in the structure built by HUNGER-RULE. Since the focus of this request is already filled, the condition of the request is true immmediately, and the request will fill the gap as soon as it is activated. This technique of using a request to specify the details of the structure built by another request is used widely throughout PAM.

The next sentence of story (1) has the following Conceptual Dependency representation:

```
((<=> ($RESTAURANT CUSTOMER HUMO RESTAURANT ORGO)))
```

That is, John played the role of the customer in doing the restaurant script at some restaurant.

PAM places this conceptualization in the input gap and examines the requests focused upon this gap. The request SUITABLE-PLAN-RULE does not have its condition met by this input, because the input is stated in the form of an action, not a plan. Thus PAM must go into bottom-up mode and find the rules applicable to this sentence.

In this case PAM finds a rule called DO-RESTAURANT-PLAN-RULE whose condition is met by the input. Once again, the question of how this rule is found is deferred until later. DO-RESTAURANT-PLAN-RULE has the following structure:

```
DO-RESTAURANT-PLAN-RULE

 Condition:     (LAMBDA (X)
                (E (PATH (<=>) X) '$RESTAURANT))

 Action:        (BUILD '((PLANBOX (*DO-$RESTAURANT-PLAN*
                                   PLANNER (NIL)
                                   RESTAURANT (NIL)
                                   )
                         ACTIONS (NIL)
                         SUBEPISODES (NIL)
                         ))
                )

 Suggestions:
  ( ((PLANBOX PLANNER) (<=> CUSTOMER) FOCUS-REQ)
    ((PLANBOX RESTAURANT) (<=> RESTAURANT) FOCUS-REQ)
    ((ACTIONS) !FOCUS! FOCUS-REQ)
  )
```

The action of this rule builds a structure denoting a plan. This structure has the following components: a planbox, a list of actions, and a list of sub-episodes. The planbox here is simply doing the restaurant script. Its planner and the restaurant used are filled in by the first two suggestions. The ACTIONS role is used to specify the actual events to which the plan gave rise. In this example, this is filled with by the third request which moves the input into this gap. The SUBEPISODES gap is used to store episodes whose goals were instrumental to the use of this plan. This slot is left empty in this story.

Having built the structure specified by request DO-RESTAURANT-PLAN-RULE, PAM places this structure into the special input gap, and checks to see if any requests focused on this gap predict the structure. This time, when PAM encounters R5 and applies its condition, it finds that the condition is met. This condition, SUITABLE-PLAN-PREDICATE, examines a list of plans associated with a goal to determine if the input is a plan for that goal. Since

*DO-$RESTAURANT-PLAN* is contained in the list of plans for the goal *S-HUNGER*, the predicate returns true. PAM then evaluates the action of R5, which moves its focus into CON12, the PLAN gap in the structure HUNGER-RULE built. Since a prediction has been satisfied, and the structures built from each input joined together, PAM stops processing this sentence. The representation created for the story contains the theme, goal, plan, and action elements that constitute the story representation shown at the beginning of this section.

The following diagram shows this schematic story representation, and the representation produced by PAM. The numbers in the representations indicate correspondences between the two. For the sake of clarity, the nodes present in the actual PAM representation have been replaced by their values.

---

Schematic story representation:

```
                  1
Theme - Hunger
  |                      2
Goal  - Satisfy-hunger
  |                            3
Plan  - Do-restaurant-script  ----> Action 4
                                   (John eat)
```

---

Actual story representation, shown by replacing nodes with their values:

```
 ________________________________________________
| This part built by HUNGER-RULE                 |
|                                                |
|                                     2          |
| ((GOAL (*S-HUNGER* PLANNER HUMO)               |
|                                           1    |
|  SOURCE (*THEME* VAL (*HUNGER-DRIVE*))         |
|             ________________________________________________
|  PLAN     | ((PLANBOX (*DO-$RESTAURANT-PLAN*              3 |
|___________|             PLANNER HUMO RESTAURANT ORGO )        |
            |                                                   |
            |   ACTIONS ( ((<=> ($RESTAURANT                    |
            |                     CUSTOMER HUMO         4       |
            |                     RESTAURANT ORGO))) )          |
            |                                                   |
            |   SUBEPISODES (NIL) )) ))                         |
            |                                                   |
            |   This part built by SUITABLE-PLAN-RULE           |
            |___________________________________________________|
```

---

In sum, the process of understanding story (1) is as follows: PAM reads in a conceptualization representating a sentence of a story, and places it in the input gap; then PAM tests the conditions of the requests focused on this gap to see if any of them predicted the input. In the sample story no such request was found, so PAM had to search for rules that might be applicable to the input.

PAM then found the request HUNGER-RULE whose condition was met by the input. HUNGER-RULE built a structure denoting an episode with the goal, theme and plan. The plan part of this structure was left empty, constituting a gap, but HUNGER-RULE made a suggestions as to how this gap may be filled. This suggestion consisted of attaching a copy of the request SUITABLE-PLAN-RULE to this gap, and focusing the request on the input gap.

Next the structure built by HUNGER-RULE was placed in the input gap to see if it met any expectations. An active request was found that was looking for an episode containing a theme. Since this request predicted this structure, PAM stopped processing and went on to the next sentence.

The next sentence was likewise not predicted by PAM. PAM had to search for rules applicable to the sentence, and found the rule DO-RESTAURANT-PLAN-RULE. This rule built a plan structure that was then placed in the input gap. The version of SUITABLE-PLAN-RULE activated by the previous sentence predicted such a plan, and inserted it in the plan gap in the story representation. This input had now led to an inference predicted by PAM, and thus processing could stop.

The understanding process in terms of rule manipulation is shown in Figure 13.

### 13.3 Request Manipulation

The last section described the general structure of requests and how PAM uses them. In this section, request structure and manipulation are examined in more detail.

#### 13.3.1 Request Chains

As was discussed previously, when a request is activated it is attached to a gap PAM is trying to fill, and focused on a gap to which its condition is to be applied. Most of the requests shown above were focused on the special gap where PAM placed input, but this need not be the case. For example, some of the requests used in suggestions were used simply to move a piece of representation from the focus into the structure built by the request making the suggestion, and were therefore focused on a filler of a role within the input conceptualization.

Figure 13
PAM's Processing Algorithm in Terms of Rule Manipulation

```
          Sentence in Conceptual Dependency
                          ||
  //==============>       ||
  ||                      \/
  ||      Predictive component:
  ||
  ||     ______________________________
  ||    |                              |
  ||    | Is there an active request   |  yes   The request adds
  ||    | focused on the input gap     | ====>  to the story
  ||    | whose condition is met by    |        representation and
  ||    | the input?                   |        the algorithm
  ||    |______________________________|        terminates successfully
  ||
  ||                      || no
  ||                      \/
  ||      Bottom-up component:
  ||     ____________________________
  ||    |                            |
  ||    | Are there rules whose      |   no   algorithm
  ||    | condition is met by the    | ====>  terminates
  ||    | input?                     |        unsuccessfully
  ||    |____________________________|
  ||
  ||                      ||
  ||                      \/
  ||     ____________________________
  ||    |                            |
  ||    | Use the rules' actions     |
  ||    | build structures and       |
  ||    | place each one in turn     |
  ||    | into the input gap.        |
  ||    |____________________________|
  ||
  ||                      ||
  ||                      \/
  ||            Inferred Explanations
  ||                      ||
  \\=================//
```

In addition, requests are often set up in such a way so that the focus of one request is the target of another. Whenever a gap is filled, PAM checks to see if any requests are focused on that gap, and test the conditions of those that are. If one of these conditions is met, PAM builds the structure specified by the action of the request, and uses it to fill the request's target. Since this is a gap that has just been filled, PAM checks to see if it has any requests focused on it, and re-iterates the cycle with these requests. In principle, the firing of one request can trigger a chain reaction of pending requests, each of which is focused on another's target.

AD-A062 629 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 5/7
UNDERSTANDING GOAL-BASED STORIES.(U)
SEP 78 R WILENSKY N00014-75-C-1111
UNCLASSIFIED RR-140 NL

4 OF 4
ADA
062629

END
DATE
FILMED
3 -79
DDC

This feature of chaining together requests is useful in a number of places. For example, one feature needed by PAM is the ability to determine if a goal has been fulfilled, and if a plan used in the service of a goal has succeeded. It is convenient to implement this ability by having a role in each plan structure that describes whether or not the plan worked. This role is called the OUTCOME, and may be filled with either a structure denoting success or one denoting failure. Thus a plan structure actually has the following format:

```
((PLANBOX (...) ACTIONS (...)
  SUBEPISODES (...) OUTCOME (...) ))
```

All requests that build plan structures make a suggestion targetted at this gap that can identify the success of that particular plan. These requests are all focused on the input gap, and test for a particular condition that marks the success of their particular plan.

In addition, each goal episode built also has an OUTCOME role. This role can be filled with a structure denoting fulfillment, abandonment or failure. Thus goal episode structures appear as follows:

```
((GOAL (...) SOURCE (...) PLAN (...) OUTCOME (...)))
```

Another suggestion is used to activate a request PLAN-SUCCESS-RULE. PLAN-SUCCESS-RULE is targetted at the outcome gap of the goal episode and focused at the outcome gap of the plan structure. This request has the following simple form:

```
PLAN-SUCCESS-RULE

 Condition:     (LAMBDA (X) (E X '*SUCCEEDED*))

 Action:        (BUILD '(*FULFILLED*))
```

That is, the condition of PLAN-SUCCESS-RULE holds if the focus is filled with a structure representing the success of a plan. The action of the request builds a structure denoting the fulfillment of a goal. In effect, PLAN-SUCCESS-RULE encodes the rule that if a plan for a goal succeeds, then the goal will be fulfilled.

The following diagram is a graphic representation of the story representation and the requests connecting its gaps:

```
              CON31 <-> Gap for goal episode outcome.

                 /\
       target    ||
                 ||

              R17  <-> Activated form of PLAN-SUCCESS-RULE,
                       looks for (*SUCCEEDED*), and builds
                 |     (*FULFILLED*) if it sees it.
        focus    |
                / \

              CON43 <-> Gap for plan outcome.

                 /\
       target    ||
                 ||

              R23  <-> Activated request that can determine
                       plan outcome, and builds (*SUCCEEDED*) if
                 |     plan works.
        focus    |
                / \

              INPUT <-> The input gap.
```

Suppose that INPUT were now filled with a conceptualization that implies the success of the character's plan. Active request R23 would find its condition met, and would cause CON43 to be filled by the structure (*SUCCEEDED*). Now the requests focused on CON43 would be tested, and the condition of R17 found to hold. The action of this request would be taken and CON31 filled with the structure (*FULFILLED*).

Thus PAM was able to fill in two gaps at once by chaining together requests. The advantage of this scheme is that it avoids testing requests unnecessarily. For example, it would be possible to have two independent requests focused at the input gap, one for targetted at the outcome gap of the goal episode and the other at that of the plan structure. However, in that case, the program would have to make two reasonably complicated tests in order to fill both gaps. Furthermore, in the case where the tests failed, the program would have to test each request separately to determine that each one's condition was not met. In contrast, by chaining the two requests together, PAM makes only one test in the case where the test fails. In the case where the test succeeds, the second test is a simple one by comparison.

In general, chaining requests together provides a way of having a large number of active requests available to build story structures at a low computational cost. Since a request is not examined until its focus is filled, only a small percentage of the active requests are usually tested at any given point. Thus the number of predicates that need be

tested at any one point is kept to a minimum, without restricting the power of those requests or significance of the action that a request can trigger off.

13.3.2 Request Types

The reader may have noticed that requests that determine plan outcome are not explanatory requests. That is, a request that determines that an event implies that a plan has succeeded does not in itself explain why that event occurred. For example, if John threated to harm Bill if he didn't give him some money, then Bill giving John the money indicates that this plan has succeeded. However, noticing that the plan has succeeded does not constitute an explanation for why Bill gave John the money; such an explanation requires inferring the goal behind Bill's action, and this is done by a rule separate from the rule that determines that John's plan has succeeded.

The problem here is that the rule that determines plan outcome is a prediction. PAM is supposed to stop processing an input when a prediction is confirmed, but this prediction might stop processing before an explanation is inferred. That is, if the explanation for the sentence requires a bottom-up inference, and the rule detecting plan success is triggered directly from the input, then PAM will stop processing before an explanation for the input has been found.

To prevent this situation from occurring, PAM flags those requests that provide explanations as being WHY requests. For example, a request that determines the outcome of a plan would not be marked as a WHY request. On the other hand a request that inferred that the goal behind Bill's giving John the money was to preserve his life would be marked as a WHY request. Thus in processing the sentence "Bill gave John the money" after hearing that John threatened Bill, the request looking for plan success will be triggered first. Since this is not a WHY request, PAM continues to process this sentence and eventually find an explanation for Bill's action that is predicted by some WHY request.

Thus an input can trigger any number of non-WHY requests without stopping the inference cycle. Once a WHY request goes off, however, the explanation cycle will terminate and PAM will go on to the next sentence. This distinction provides a way of doing non-explanation functions in PAM without interfering with the basic flow of control in the explanation cycle.

Another problem that arises with requests involves communication between a request and the suggestions it makes. For example, in the previous section I showed SUITABLE-PLAN-RULE, a request for determining if an input could be a plan for a goal. This request was given as follows:

SUITABLE-PLAN-RULE

Condition: SUITABLE-PLAN-PREDICATE

Action: !FOCUS!

The problem is that SUITABLE-PLAN-PREDICATE needs some way to refer to the goal for which it is trying to find a suitable plan. That is, this request is targetted at a gap in a structure that contains a goal, but has no way to reference the goal in that structure when it is trying to determine if its condition has been met.

To solve this problem, PAM marks requests that build certain important structures as being FOUNDERs. The name of the last FOUNDER built in the course of activating requests is kept around and can be referenced by all those requests activated by the FOUNDER, or by the suggestions of a FOUNDER, and so on. To reference the structure built by the FOUNDing request, the special symbol !PARENT! is used. For example, SUITABLE-PLAN-RULE actually has the following form:

```
SUITABLE-PLAN-RULE

    Condition:    (LAMBDA (X)
                   (SUITABLE-PLAN-PREDICATE
                    (PATH '(GOAL) !PARENT!) X))

    Action:       !FOCUS!
```

SUITABLE-PLAN-PREDICATE is really a function of two arguments, a goal and a plan. The goal is specified by a PATH through the !PARENT!, or the structure built by the FOUNDing request. This structure is a goal episode and the desired goal fills one of its roles. When this request is activated, PAM replaces the symobl !PARENT! with the name of the actual node whose value is the structure built by the FOUNDING request. Thus when the condition of the request is eventually tested, the first argument will always evaluate to the desired goal.

FOUNDERs are almost always goal episodes. A request activated by a request that builds a goal episode can activate another request, and as long as a new FOUNDER request is not activated, the new requests can all use !PARENT! to refer back to the FOUNDing request. In addition, the use of !PARENT! within a request that is a FOUNDER will reference the structure built by the last FOUNDing request. Thus a request that is loaded through several levels of suggestions can still reference a structure built by a request many levels before.

### 13.3.3 Removing Requests

Up until now we have been concerned mostly with the activation and use of requests. Another important feature of PAM is its ability to remove requests. Removing requests is important for two reasons. First, a request that has outlived its usefulness may find its condition met by chance later on in the story, and go off unexpectedly, filling some gap with an inappropriate value. Second, even if there is no chance that the request will go off, the program may still waste time checking the request's condition whenever the gap which it is focused at is filled.

Removing a request means that the request name is taken out of the list of requests under the "Focused-requests" property of the requests focus. The request will then no longer be tested when that gap is filled. PAM removes requests under the following circumstances:

1. Any request that has fired off (i.e., its condition has been confirmed, its action taken and its suggestions activated) is removed.

2. If a request fills a gap, then all the requests targetted at that gap are removed.

3. If a request fills a gap that will not be filled again, then all the requests focused on that gap are removed after they are tested.

Requests targetted at a gap are removed when the gap is filled because there is no longer anything left for them to do. The requests were targetted at the gap in the first place in order to fill it, and now that it has been filled, they serve no useful purpose. For example, upon encountering a planbox like ASK or BARGAIN that requires a response from someone, PAM sets up one request predicting compliance, and another predicting denial. Both requests are targetted at a gap for the RESPONSE role of the plan. If one of the requests is confirmed and the RESPONSE gap filled, then both requests are removed. Thus this technique provides a way of removing requests whose purpose no longer exists even if the request was not used.

To find the requests targetted at a gap, PAM maintains a list of such requests under a property of each gap. When a gap is filled, then PAM goes through this list of targetted requests and removes each one in turn.

Requests focused on a gap are removed when the gap is filled since these gaps will not change again. This is true of all the gaps in PAM, with the exception of the input gap. This gap is filled many times throughout a story, and it would be undesirable to remove predictions that might applicable to future sentences whenever the gap were filled. When a gap other than the input gap is filled, however, it is filled for good. All the requests focused on it tested, and those requests that did not fire will never be tested again. These requests can be removed for convenience since they no longer serve any function.

Suggestions, chains of requests, and request removal are all means of organizing rules so that the appropriate rule is activated at the right time, without having many unnecessary rules around also. One additional mechanism use for this purpose involves attaching requests to target gaps. PAM has a number of rules that suggest attaching requests to target gaps without checking to see if the gap is in fact filled. This corresponds to a situation in which PAM has a number of heuristics for filling a gap but does not know which one is applicable in the present case. Thus one request may have already filled a gap using some heuristic, and later on another request may try to fill the gap again using another heuristic. This situation is desirable because in a very similar situation the first request may be unable to fill the gap, and PAM may depend on the second request to do the job.

If a suggestion is made to attach a request to a target gap, but that gap is already filled, PAM simply refuses to heed the suggestion. Thus suggestions are in fact just that, ideas about where to find a filler that PAM may chose to ignore under some circumstances. If a request has already filled the gap, the suggestion is ignored, but if the gap is empty, PAM will activate the suggestion to try to fill it. In this manner, a number of heuristics for gap filling may be scattered throughout a set of rules without concern for overlap - an attempt to do something already done by a previous heuristic is harmless.

Thus PAM engages in request access, request activation, request testing and request removal. The flow of control in this process is summarized in Figure 14.

## 13.4 The bottom-up mechanism

If PAM has no active requests that predict an input, it must try to find rules whose condition is met by the input. For example, suppose PAM sees an event of the form

(2) John told Bill he would give him five dollars if he gave him his bicycle.

PAM needs to find a rule that will interpret this sentence as an instance of bargaining. If no active request can interpret the sentence, PAM must located an inactive request that can. That is, PAM must find a rule whose condition is met by this sentence, and whose action will be to build a structure denoting a BARGAIN-OBJECT planbox. Then the predictive component can be used again to see if there is any gap in the story representation to which this plan may be attached.

**Figure 14**
**Request Testing and Activation Sequence**

```
When a gap changes:

         ________________________
        |                        |  no
        | Are there requests     | ====> Finished
        | focussed on the gap?   |
        |________________________|

                   ||
                   || yes
                   \/
For each request
         ________________________
        |                        |  no
        | Is the condition of    | ====> next request
        | the request true?      |
        |________________________|

                   ||
                   || yes
                   \/
         ______________________________________
        |                                      |
        | Fill attached gap by action of request. |
        |______________________________________|

                   ||
                   ||
                   \/
         ______________________________________
        |                                      |
        |     Activate suggested requests.     |
        |______________________________________|

                   ||
                   ||
                   \/
         ______________________________________
        |                                      |
        | Remove requests attached to filled gap. |
        |______________________________________|

                   ||
                   ||
                   \/
         ______________________________________
        |                                      |
        | Test gap filled by action of request. |
        |______________________________________|
```

The primary difficulty here is that we need to find such a rule amidst a haystack of others. The PAM solution to this problem uses a discrimination net to index requests (Feigenbaum, 1961). PAM's discrimination net is a binary tree with predicates at each non-terminal node. The net is applied to an input as follows: The predicate at the root of the tree is applied to the input, and depending on the outcome, the "true" sub-tree or the "false" sub-tree will be chosen. This procedure is iterated using the chosen sub-tree until a leaf (terminal node) is reached. This node is then returned.

The leaves of PAM's discrimination net are the names of requests. Each predicate in the tree is part of a predicate of some request, so that when a leaf of the tree is reached, the conjunction of predicates tested on the way is equivalent to the condition of the requests at that leaf.

For example, when the conceptualization underlying sentence (2) above is dropped through PAM's discrimination net, the following series of predicates is evaluated:

```
CONCEPT IS

((ACTOR HUM0 <=> (*MTRANS*)
  MOBJECT ((CON ((ACTOR HUM1 <=> (*ATRANS*) OBJECT PHYS0
                 TO HUM0))
              LEADTO ((ACTOR HUM0 <=> (*ATRANS*) OBJECT MONEY0
                       TO HUM1))
           ))
  TO (*CP* PART HUM1)
))
```

```
PREDICATE TRACE:

   (HAS (MODE) *NEG*)                                  FALSE
   (HAS (MODE) *CAN*)                                  FALSE
   (EQU (<=>) NIL)                                     FALSE
   (EQU (<=>) *ATRANS*)                                FALSE
   (EQU (<=>) *MTRANS*)                                TRUE
   (SAMETOK (MOBJECT IS PART) (ACTOR))                 FALSE
   (EQU (MOBJECT IS) *GOAL*))                          FALSE
   (EQU (MOBJECT CON <=>) NIL))                        FALSE
   (SAMETOK (MOBJECT CON ACTOR) (TO ACTOR))            TRUE
   (HAS (MOBJECT CON MODE) *NEG*))                     FALSE
   (SAMETOK (MOBJECT CAUSAL ACTOR) (ACTOR))            TRUE
   (EQU (MOBJECT CAUSAL <=>) NIL)                      FALSE
   (EQU (MOBJECT CAUSAL <=>) *ATRANS*)                 TRUE


REQUEST CLUSTER FOUND : BARGAIN-OBJECT-CLUSTER
```

Each predicate tests a piece of the conceptual structure specified by a path. For example, the path (MOBJECT CON ACTOR) says to first look at the conceptualization in the MOBJECT slot of the main

conceptualization, then to look in its CON slot, and finally, to look in the ACTOR slot of that conceptualization. This slot is filled with the token HUM1, which points to a representation for Bill.

The predicates in the above example first ask if the conceptualization is a negative, and then if it is potential, both of which are false. It then asks if it is an action, and when this is confirmed, what kind of action. Finding that it is an MTRANS, or transfer of communication, the program then asks if it is within the same individual (remembering would be an instance of such an event). Since it is not, various other MTRANS possibilities are tried, until it is established that the MTRANS is to the same person specified in the causing part of the concept transferred, and that his doing the stated action will lead to the person making the MTRANS to give him something. At this point, a leaf of the tree is reached. This leaf is called BARGAIN-OBJECT-CLUSTER, and it contains a list of one request - a request whose action will build a BARGAIN-OBJECT planbox.

As described above, when a bottom-up request is used to build a structure, that structure is placed in the input gap, and the requests focused on the gap are examined. If a request predicts the structure, PAM can stop. Otherwise the cycle is reiterated. For example, suppose that no request predicted the BARGAIN-OBJECT structure just described. In this case, this structure would be thrown back into the discrimination net, which would return requests that would explain the bargaining. For example, one such request might build a structure denoting a change of possession goal. If no active request predicted this structure, it too would be thrown back down the net. This time, a request would be returned that stated that this goal was instrumental to a plan to use the bicycle. If a request existed that predicted a plan for changing one's location, its condition would now be met and the explanation cycle could cease.

### 13.4.1 Testing

When PAM finds a request in bottom-up fashion, it must go through a test phase before it can actually use the request. PAM will occasionally retrieve more than one rule when using the discrimination net, and each rule may offer a different interpretation of the input. Eventually a decision is made as to which rule to use because the structure produced by one rule will be predicted by some request, but that produced by another will not. Since the suggestions specified by some of these rules may change the story representation is various ways, it is desirable to find the correct rules to use before the actions the rules dictate are actually taken.

To test a rule, PAM activates the requests it finds via the discrimination net by focusing them on the input gap and targetting them on a dummy gap. That is, PAM builds a copy of the request just as it does when it obeys a suggestion. Then PAM performs the normal evaluation of the action and suggestions of this rule, with the following exceptions: PAM does not activate suggestions that would change the story representation outside of the structure built by a

request. Thus no global changes are made. Also, the requests activated by these suggestions are kept separate from the requests already activated.

When PAM finds a sequence of rules that can explain the input, it first removes any requests activated in the test phase that are still present. Then it activates each rule that it used in the explanation sequence, making the target of one rule the focus of the next, and so on. Eventually a chain of requests is activated starting at the input gap and ending with some prediction already in the story representation. The input gap is then tested again in the normal manner. Since a request just activated is now focused on the input gap and has a condition met by the input, a request chain is fired off. This time, all the suggestions made by the requests are heeded, and the actual story representation is altered.

## 13.5 Efficiency of the model

There are two parts of PAM that are potential trouble spots in terms of processing time. First there is access of requests via the discrimination net. As the number of requests added to PAM increases, the time required to fetch them will also increase. In actuality, this problem is not significant for two reasons. One, using a discrimination net guarantees that the time required to find a request increases logarithmically with the number of requests in the system. This is slow enough to assume that a large increase in the number of requests will not substantially increase processing time. In addition, in comparison to the time required by the rest of the program, the time required to access a request is trivial. PAM spends almost all its time building the structures and copying over requests, and spends a little over 3 percent of its time accessing requests (More detailed statistics are presented toward the end of this chapter). It is estimated that each doubling of the number of requests in PAM's discrimination net increases processing time by about 10 milliseconds. Thus the access of rules does not appear to be problematic.

A second source of slow-down is more difficult to put aside. When PAM looks for requests, it sometimes returns with more than one. Each of the structures indicated by the actions of these requests must be tested, and if none is predicted by another request, each structure must be fed into the discrimination net again. If these structures each caused several more requests to be found, and if this cycle continued for long enough, the program could bog down building hypothetical explanations - get lost searching explanation space, as it were.

This situation has never occurred, and there are a number of reasons why it is unlikely to. First, it is rare that many explanations should be stored under the same concept. In practice, two is unusual and three almost unheard of. Second, there are usually not many levels between the input and the story representation. The cycle of repeated explanations does not go on for arbitrary numbers of iterations; seven was the length of longest chain PAM needed for any story it understood. Stories whose sentences are a long conceptual distance away from one

another would be difficult for humans to understand anyway, and I do not expect my program to out-perform people at this task.

Thus PAM's algorithm seems reasonably efficient. Rule accessing is not a problem, and finding too many potential explanations does not happen in practice. However, the ultimate test of these conjectures will not occur until enough rules have been added to the program to cover a large percentage of natural language utterances.

## 13.6 Pronouns

PAM finds the correct references for all the pronouns used in the stories shown in this thesis. PAM makes these references without any special reference heuristics. Instead, these references are resolved as a by-product of the other processes PAM must perform anyway to understand the story.

As PAM processes a story, it evaluates many predicates, including those in its discrimination net, and those in the conditions of requests it may test. Often these predicates ask to compare a piece of input or inferred conceptualization to a piece of story representation. Whenever the structures being compared are picture producers (representations for objects, people or places), PAM does the comparison via a special routine. This routine states that the items being compared are the same if the new one <u>could be</u> the old one. That is, if the routine is matching the representation for "John" against the representation for "he", it will decide that they are the same since "he" could be "John". This routine then puts the pair on an association list for future reference. When the sentence being processed in eventually understood, PAM goes through the structures built up and replaces each item on the pronoun association list with its referent.

For example, recall story (2) above:

(3) John wanted money. He got a gun and walked into a liquor store. He told the owner he wanted some money.

As was mentioned previously, after reading these sentences, PAM has made a prediction that if the owner gives John the money, he did so in response to John's threat. Suppose the next sentence were "He gave him the money." The condition of the request predicting compliance must check to see if the actor of this conceptualization is the owner, and the recipient John. Since the condition uses the special comparison routine, it will compare the representation for the first "he" in the sentence with that for the owner, and the second with that for John. These both match, and the condition is found to hold. Thus references to the first "he" will be replaced by references to the owner, and references to the second by references to John. This scheme is similar to that used by Charniak (1972) and Cullingford (1978).

While I cannot make claims for this algorithm beyond the actual stories PAM uses it to process, it does raise an interesting possibility. Namely, it is possible that all referential problems are

solved as a by-product of processes whose purpose is to find the connection between the sentences of a text. That is, rather than have a set of rules for the sake of resolving references, it may be that the need for all such rules is subsumed by understanding rules that are necessary with or without the presence of pronouns in the text.

In this view, there is no such thing as the reference problem. Rather, there is only the problem of establishing connections between sentences. Whether all reference problems will fall out of other understanding problems remains to be seen. However, if the stories PAM processes are any indication, it seems that the existence of a separate, definable problem of reference is in doubt.

## 13.7 Odds and Ends

### 13.7.1 Listnodes

Some of the roles in a story representation can have more than one filler. For example, a plan structure contains a role called SUBEPISODES that is used to point to episodes whose goals are instrumental to the performance of the plan. There may be several such episodes in a story, so the SUBEPISODES role must be able to accommodate them all. The plan role of a goal episode was shown in the preceding sections to contain a single filler, but of course, a story may contain more than one plan for a goal if the first attempts fail. Thus this role must also be able to accommodate multiple fillers.

For this purpose, PAM has a special kind of node called a listnode. Listnodes can be filled with any number of structures. When a request is triggered whose target is a listnode, PAM adds the structure built by the request on to the end of current value of the listnode. Likewise, when PAM is trying to attach a request to a node, it will attach it to a listnode even if there is a structure already present in the node.

While listnodes are an obvious means to allow a gap to be filled with more than one filler, they present some problems for request management. For example, listnodes can be filled by more than one request, so it is not possible to remove all requests attached to a list node when one of the requests fills the gap. This causes a problem in the following case. Along with each event, PAM stores the episodes which that event initiated. If the event is part of a plan that involves a response by another character, like ASK, BARGAIN, or THREATEN, then the event initiates the episode constituting the response. In general, since an event may give rise to any number of episodes, the INITIATES role in the PAM representation of an event must be a listnode.

In processing a story involving one of these plans, PAM will activate a number of requests to explain the possible responses. For example, an ASK planbox activates one request that looks for compliance, and another that looks for denial. These requests are conceptually in opposition, i. e., one firing precludes the firing of the other. However, these requests are for filling in the INITIATES listnode of the

current event. If the requests were attached to this node, then one of them firing would not cause the other to be removed. This situation is potentially damaging because the request that was not fired may be triggered accidently later on. For example, if the request looking for denial were triggered, and then later on the planner tried another plan that worked, the request looking for compliance to the previous plan may be triggered and attach the compliance episode to the wrong plan.

To resolve this problem, PAM has to set up a dummy node and attach it to the listnode for the INITIATES node via FOCUS-REQ. Recall that FOCUS-REQ just passes along its focus to its target once the focus is filled. This copy of FOCUS-REQ is targetted at the listnode and focused on the dummy node. The requests for compliance and denial are targeted on the dummy node, and focused on the input gap. That is, we have interposed a dummy node between the two requests and the actual gap we desire them to fill. This is diagrammed as follows:

---

```
                    CON40     <->  INITIATES listnode
                     /\
                     ||
                     ||
                     ||
                    R20       <->  copy of FOCUS-REQ
                     ||
                     ||
                     ||
                     / \
                    CON30     <->  copy of FOCUS-REQ
                  /\     /\
                  ||     ||
copy of           ||     ||        copy of
denial    <->    R20    R21   <->  compliance
request           ||     ||        request
                  ||     ||
                  ||     ||
                  / \  / \
                   INPUT <-> the input gap
```

---

When one of these requests is triggered, it fills the dummy gap, and because the dummy gap is not a listnode, all the requests attached to the node are removed. Because this gap has just been filled, the requests focused on it are tested, and the copy of FOCUS-REQ will add the structure in the dummy node to the INITIATES node. Since this copy of FOCUS-REQ is focused at a gap that will not be filled again, it too is removed. Thus the INITIATES node will be filled and no unwanted requests remain active.

### 13.7.2 Side-effects

Recall that the action field of a request always builds a structure, and has no side-effects. There are a number of cases, however, in which side-effects are useful. For this purpose, each request in PAM has two additional fields, SIDE-EFFECT and MESSAGE.

SIDE-EFFECT is a list of arbitrary LISP expressions that are evaluated after the action of a request is taken. SIDE-EFFECTs are generally used for bookkeeping purposes. For example, if PAM learns that John loves Mary, it is often convenient for future accessing to store this information under the representation for John and the representation for Mary. This is done by putting a side-effect on the request that does the goal-based processing of this sentence.

A special case of side-effects are messages that PAM prints as it makes inferences during understanding. Messages are placed under the MESSAGE field of a request, and cause PAM to express these messages when the request fires off. Both SIDE-EFFECTs and MESSAGEs are ignored during the test phase to insure that no global effects occur and that false messages are not printed out prematurely.

13.8 Facts about PAM

PAM is written in UCILISP and MLISP on a DEC System 20/50. The program currently runs in 177 pages (90k words). This includes 40k words of free storage used for building story representations and copies of requests. The question-answerer and English generator fit into this space, but ELI, the English Language Interpreter, does not.

The English generator used to tell the story from different viewpoints is called PHLUENT (PHrasily flUENT generator), and was written by Rod McGuire and Rick Granger. This program is also used to answers to questions in English. The question answering routines were written by Janet Kolodner. Michael Lebowitz helped to extend PAM in its later stages.

It takes PAM an average of 5.58 CPU seconds to process a sentence. The processing time per sentence ranges from .9 to 15.2 seconds. This figure does not include the time required to analyze an English sentences into its Conceptual Dependency representation. ELI averages about 3-4 CPU seconds to perform this task.

Little attempt has been made to make PAM as fast or as small as possible. There are a number of straightforward modifications that could reduce the above figures somewhat. For example, all of PAM's code is interpreted. Compiling this code would probably cause a speed-up of about a factor of three. This figure is based on the speed-up that occurred when ELI's code was compiled. ELI uses requests that are similar to those PAM uses, so it seems reasonable to expect a comparable speed-up. This increase in speed is somewhat smaller than the usual advantage one obtains from compiling UCILISP. Normally compiled code is about eight to ten times as fast, but since requests are still interpreted, the full advantage of compiling is not realized.

The total size of the program could be reduced by garbage collecting requests that have been removed. PAM currently saves these requests for debugging purposes. Also, the structures build during the test-phase (the part of the processing where PAM is making bottom-up inferences from an input, but has not yet connected them to the story

representation) are saved, but they too could be thrown away. Together, these objects consume about three-quarters of the free storage space used by PAM.

PAM's discrimination net currently uses 174 predicates. These distinguish among 75 request clusters. That is, there are 75 leaves on the tree PAM uses to find requests applicable to a conceptualization. The tree is as sparse as it is because many of the predicates were written in anticipation of adding new requests. In fact, upwards of 90 percent of the "dead-end" leaves of the tree correspond to meaningful conceptualizations and therefore will eventually have requests attached to them.

It takes 108.75 milliseconds on the average to access a rule using the discrimination net. The bottom-up cycle that performs this access iterates 1.6 times per sentence, averaged over all the stories PAM has processed. Thus PAM spends about 3.12 percent of its time accessing rules via the discrimination net.

This figure is significant because accessing time using a discrimination net grows logarithmically with the number of leaves of the tree, provided the tree grows evenly. The average number of predicates tested to reach a leaf is 10.83, with a standard deviation of 2.375. Thus the average execution time of each predicate is about 10 milliseconds. If the net keeps growing evenly, and new predicates cost about the same as old ones same, then each doubling of the number of requests accessible by the net increases processing time only 10 milliseconds.

On the average, each sentence processed by PAM causes 40.73 requests to be activated. This number is as large as it is because (1) it includes all the requests from the test phase as well as the those used to actually change the story representation, and (2) many of these requests are utility requests like FOCUS-REQ that just contribute small pieces to a larger structure built by a more meaningful request. FOCUS-REQ alone is activated 19.6 times per sentence, constituting half the requests activated. Discounting the redundancy of the test phase and taking into account the purpose of the requests, I estimate that about 5-7 "meaningful" requests are activated for each sentence.

The number of active predictions averages 4.79. This is the number of requests focused on the input gap that are tested each time a sentence is read. This figure tends to vary in the range of 0 to 8.

PAM currently understands 16 different classes of stories, ranging in length from 2 to 9 sentences. A class is distinguished by the kinds of inference processes that are needed to understand the stories in that class. For example, one class of stories involves inferring one character's goal bottom-up from a description of an action. Other classes include the more complicated goal relationship situations like goal subsumption termination (see Chapter 5) and goal conflict leading to goal abandonment (Chapter 7). In each case, the stories in one class differ from those in another by more than a substitution of one plan or goal for another. For example, the story

John wanted a book. He asked Bill for one.

is in the same class as

John wanted a book. He told Bill he would buy one from him for a dollar.

In addition to the rules needed to understand the computer examples shown in this thesis, PAM also contains rules for other stories in the same class. Most of these rules, like those for identifying most planboxes, were not listed in this thesis. Of those rules specified in the thesis, PAM actually contains about one third.

### 13.9 Summary

A snapshot of PAM as it is processing a story shows a representation filled with gaps, and requests attached to these gaps waiting for events to trigger them. When a new input occurs, PAM tests the requests awaiting input, and if one of them has a condition that is true, it adds it to the story representation. The request may also make suggestions that other requests be activated.

If no request has its condition made true by the input, PAM drops the input down a discrimination net to find new requests that can explain it. If one is found, its action is taken, and the explanation it builds is placed in the input gap. The requests waiting for input have their conditions tested again. The cycle continues until a request waiting for input finally sees its prediction confirmed, or until PAM can no longer infer explanations from the input.

# CHAPTER 14

## A DETAILED EXAMPLE

This chapter provides a detailed examination of a trace of a PAM story understanding session. A simple story is used to illustrate the processing discussed in the previous chapter, as well as to point out some additional technical difficulties.

```
[PHOTO:  Recording initiated  Sat 5-Aug-78 2:46PM]

@RUN PAM

*(UNDERSTAND CD3)

THE STORY IS

 JOHN WAS LOST.
 HE PULLED OVER TO A FARMER STANDING BY THE SIDE OF THE ROAD.
 HE ASKED HIM WHERE HE WAS.
```

| COMPUTER OUTPUT | ANNOTATION |
|---|---|
| PROCESSING ... | |
| NEXT INPUT IS:<br>(JOHN WAS LOST) | |
| CONCEPTUALIZATION IS:<br>((CON<br>((ACTOR HUM0 IS (*PROX* PART<br>(*UNSPEC* CLASS (*LOCATION*))))<br>TIME (FORM11))<br>IS (*MLOC* VAL (*LTM* PART HUM0)))<br>MODE (FORM12) TIME (FORM13)) | The first sentence in Conceptual Dependency denotes that John did not know his own location, i. e., "The concept that HUM0 was at some location was not in HUM0's memory." |

HUMO is a <u>token</u>, that is, a permanent memory referent created for the character John. Information about tokens is kept on the property list of the atom denoting the individual. For example, HUMO has the following properties:

```
HUMO

 TOKEN:        T

 CLASS:        (*PERSON*)

 PERSNAME:     (John)

 GENDER:       (*MASC*)
```

Thus, HUMO is named John and is a masculine member of the class of persons.

Now PAM examines the requests attached to the input gap to see if any of their conditions have been met by this input. The only requests around at the beginning of a story are looking for goal episodes and static information (e. g., "John lived in New York"), and do not fire off upon seeing this input. Thus PAM embarks upon the bottom-up phase and tries to find a rule applicable to this input.

---

```
NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
FIND-OUT-REQ

EXPLANATION IS GOAL:
(*DKNOW* PLANNER HUMO RECIPIENT
 HUMO FACT ((ACTOR HUMO IS (*PROX*
 PART (*UNSPEC* CLASS
  (*LOCATION*)))) TIME (TIMK2)))

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH
```

PAM drops the input down its discrimination net and finds the request FIND-OUT-REQ. FIND-OUT-REQ encodes the knowledge that a person who doesn't know where he is will want to find out where he is. Thus FIND-OUT-REQ builds the goal structure shown.

---

FIND-OUT-REQ is written as follows:

```
        FIND-OUT-REQ

TEST    T

ACTION  (BUILD '#DKNOW-EPISODE)

SUGGESTIONS
        ( ((GOALFORM GOAL PLANNER) (IS VAL PART) FOCUS-REQ)
          ((GOALFORM GOAL RECIPIENT) (IS VAL PART) FOCUS-REQ)
          ((GOALFORM GOAL FACT) (CON) FOCUS-REQ)
          ((ATTEMPTS) FIND-OUT-AT-REQ)
        )

MESSAGE (PRTGOAL (PATH '(GOALFORM GOAL) !STRUCT!]

FOUNDER T
```

The condition of FIND-OUT-REQ is T, indicating that its actual predicate is distributed throughout the tests in the discrimination net. The action of the request builds a structure called #DKNOW-EPISODE, which is a structure denoting a goal episode with a Delta-know goal. The suggestions of FIND-OUT-REQ describe how to fill out this structure from the particulars in the input. The message field says to express the actual goal built in pseudo-English. The request is also marked as being a FOUNDER, meaning that the structure it builds can be referenced by the requests it activates by its suggestions.

Upon finding FIND-OUT-REQ, PAM makes a copy of it, and focuses the copy on the input. The copy is targeted at a dummy gap PAM creates for this purpose, CON39. In making the copy, PAM replaces symbolic references in FIND-OUT-REQ with their actual referents. For example, the MESSAGE field of FIND-OUT-REQ references the special symbol !STRUCT!. This symbol refers to the structure that the request itself will build. PAM replaces this symbol with the name of a gap. When PAM actually builds the structure specified by the request, it will move the structure into this gap. Since the actual name of this gap appears in the message field, when the message is evaluated, its code will reference the goal within the structure build by the correct copy of FIND-OUT-REQ. Replacing these symbolic references by their referents at the time of activation is a way of remembering the correct context of the request when its condition is tested or its action evaluated, i. e., it effects a closure of the request in the context in which it was created.

PAM now examines this copy of FIND-OUT-REQ. Since its test is always true, PAM evaluates the function in the action field, building an instance of #DKNOW-EPISODE. This episode has the following form:

```
#DKNOW-EPISODE:

    (*GOAL-EPISODE*

    GOALFORM ((GOAL (*DKNOW* PLANNER (NIL)
                             RECIPIENT (NIL)
                             FACT (NIL))
                      SOURCE (NIL)
                      RELATIONS (*LIST*)
                      OUTCOME (NIL) ))

    ATTEMPTS (*LIST*))
```

This goal episode structure is actually somewhat more complicated than the goal episode structure I alleged PAM used in the beginning of this chapter. It has two roles, a GOAL-FORM role and an ATTEMPTS role. The GOAL-FORM role is filled by a structure that has roles for the actual goal, its source, its outcome, and its relations to other goals (e. g., goal competition, goal conflict, etc.). The ATTEMPTS role is a listnode, which is designated by the special form *LIST*. This gap can be filled with a number of attempts aimed at fulfilling this goal.

An <u>attempt</u> is a structure that contains a plan role plus some additional information about that plan. For example, FIND-OUT-REQ above places an attempt structure into the ATTEMPTS gap of the goal episode it builds by targetting the request FIND-OUT-AT-REQ at this gap. This request has the following form:

```
        FIND-OUT-AT-REQ

TEST    T

ACTION  (BUILD '#DKNOWAT)

SUGGESTIONS
        ( ((SOURCE) !PARENT! FOCUS-REQ)
          (!PARENT! (GOALFORM OUTCOME) !STRUCT!
           (OUTCOME) GSUCCEED-REQ)
          ((OUTCOME) !INPUT! RESULT-IN-DKNOW-REQ)
          ((PLAN) !IN! SUITABLE-PLAN-REQ)
        )
```

That is, FIND-OUT-REQ has a suggestion that causes PAM to make a copy of FIND-OUT-AT-REQ and attach it to the ATTEMPTS gap in the structure built by FIND-OUT-REQ. Since FIND-OUT-AT-REQ has a test that is always true, it goes off immediately, and builds an instance of the structure called #DKNOW-AT. This structure is as follows:

```
        #DKNOW-AT:

            ((PLAN (NIL) SOURCE (NIL) OUTCOME (NIL]
```

The PLAN role of an attempt is used to contain the actual plan. The outcome holds a structure denoting the success or failure of the attempt, and the source is simply a backpointer to the goal episode to which the attempt is attached.

DKNOW-AT-REQ attaches a number of requests to this structure. For example, the first suggestion of DKNOW-AT-REQ fills the source of the attempt with a pointer to the entire goal episode (Recall that !PARENT! is used to refer back to the structure built by the last FOUNDER request). The second suggestions focuses the request GSUCCEED-REQ on the outcome gap of the attempt, and targets it at the outcome gap of the goal episode. This request will build a structure denoting goal fulfillment if it sees a structure denoting plan success. The third suggestion looks for an event that implies the attempt succeeded, and that fourth suggestion activates SUITABLE-PLAN-REQ. This is the request referred to in the previous chapter that looks in the input for an appropriate plan for the goal in the goal episode.

The general form of a suggestion is following:

(!SPNAME1! (A B C) !SPNAME2! (X Y Z) REQUEST-NAME)

This states that the request REQUEST-NAME should be activated. The target of the request is the value of (PATH '(A B C) !SPNAME1!), where !SPNAME1! is a special symbol used in a request to refer to some structure. If no name is specified, PAM uses the default !STRUCT!, denoting the structure built by the request making the suggestion. If the path specifier (A B C) is not given, PAM assumes that the target is specified by the special symbol.

The focus of the request is the value of (PATH '(X Y Z) !SPNAME2!). If !SPNAME2! is not specified, PAM uses the default !FOCUS!, denoting the focus of the request making the suggestion. If no path is specified, the special symbol is taken as the focus.

Note that the suggestions of the request FIND-OUT-AT-REQ refers to !IN! and !INPUT! respectively. PAM actually keeps two input gaps, one for the actual input, and one for inferences generated from the input during the bottom-up phase. This distinction is not strictly necessary, but in practice, requests that are triggered by an inference are not also triggered by actual inputs. For example, it happens that no input corresponds exactly to the form that SUITABLE-PLAN-REQ looks for, and no inference will trigger RESULT-IN-DKNOW-REQ. Thus it is slightly more efficient to maintain two gaps for these purposes. PAM uses the gap IN for this purpose, and refers to it symbolically by the name !IN!.

So far, PAM has built a goal episode and filled in an attempt and a goal. It has also connected the two outcome gaps with a request, and activated a request looking in the input for a plan for the goal. Another request has been activated looking for the success of the attempt.

PAM then moves this structure into IN, the gap for new inferences, and examines the requests focused on that gap. None of these requests fire, so PAM now looks for a rule to explain this goal episode:

```
TESTING EXPLANATION OFFERED BY
UNK-DPROX-REQ

EXPLANATION IS GOAL:
(*DPROX* PLANNER HUMO OBJECT HUMO
 LOCATION (NIL))
 PREDICTED PLANS  NIL

NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH
```

PAM drops the structure built by FIND-OUT-REQ down the discrimination net and finds the request UNK-DPROX-REQ. UNK-DPROX-REQ encodes the rule that a person who wants to know where he is is probably trying to go someplace, and that finding out where he is is instrumental to this goal. Thus the request builds a goal episode denoting that John wants to go somewhere.

UNK-DPROX-REQ is copied, tested and its structure built. This structure, CON64, is placed in IN, and not found to be wanted by any request. Thus PAM iterates the cycle once again.

```
TESTING EXPLANATION OFFERED BY
LOC-SUBGOAL-REQ

EXPLANATION IS GOAL:
(*DPROX* PLANNER HUMO OBJECT HUMO
 LOCATION (NIL))
 PREDICTED PLANS  NIL

EXPLANATION CONFIRMS PREDICTION
INIT-REQ

FOUND EXPLANATION SEQUENCE:

FIND-OUT-REQ -> UNK-DPROX-REQ
 -> LOC-SUBGOAL-REQ
```

The request LOC-SUBGOAL-REQ is found. This request encodes the information that being someplace is often instrumental to some plans, but that the particular plan is not known. The request modifies the previous goal episode to record this information.

Now when the episode built by LOC-SUBGOAL-REQ is placed in IN, the condition of INIT-REQ is met. INIT-REQ is a request activated at the beginning of PAM, and which looks for a goal episode whose source is a theme, or an unknown plan. Since the latter case holds for the goal structure above, INIT-REQ fires.

Finding a WHY request that is triggered by an input means that an explanation for the input has been found. INIT-REQ is a WHY request, so the bottom-up phase is now complete. PAM removes all requests that have become active in the bottom-up phase, and restores its memory to the state it was in before the phase began. The entire result of this phase is the explanation sequence produced. This sequence is interpreted as follows: Activate the request FIND-OUT-REQ, focus it at the input gap,

and target it at a dummy gap. Then activate the request UNK-DPROX-REQ, focus it at the dummy gap, and target it at a new dummy gap. Activate LOC-SUBGOAL-REQ, focus it on the last dummy gap, and target it at CON4. CON4 is the gap used to deposit inferences.

The actions of these requests are now allowed to change the real story representation:

| | |
|---|---|
| *** ADDING TO STORY REPRESENTATION: | |
| LOADING PREDICTION RESULT-IN-DKNOW-REQ | Firing off the requests just found causes new predictions to be made. These are discussed below. |
| LOADING PREDICTION SUITABLE-PLAN-REQ | |
| INFERRED GOAL: (*DKNOW* PLANNER HUMO RECIPIENT HUMO FACT ((ACTOR HUMO IS (*PROX* PART (*UNSPEC* CLASS (*LOCATION*)))) TIME (TIMK2))) | These are the goals PAM inferred while trying to explain the input. |
| INFERRED GOAL: (*DPROX* PLANNER HUMO OBJECT HUMO LOCATION (NIL)) | |

That is, PAM tests the requests added, and FIND-OUT-REQ fires and produces a goal episode denoting a Delta-know goal. This is placed in a dummy gap and causes UNK-DPROX-REQ to fire. This request produces a Delta-prox episode to which the first episode was instrumental. This structure is placed in a dummy gap, and caused LOC-SUBGOAL-REQ to fire. The structure built now has an unknown plan for its source, and is placed in CON4. Filling this gap causes R0 to fire. R0 moves the entire structure built into a gap used to contain entire story representations. In this phase the requests are being used to change the actual representation, so PAM also fires off their side-effects and message fields. There are no side-effects in these requests, but the message fields were responsible for the "INFERRED goal" messages above.

PAM now reads the next sentence:

```
NEXT INPUT IS:
 (HE PULLED OVER TO A FARMER
  STANDING BY THE ROAD)

CONCEPTUALIZATION IS:
((ACTOR HUM0 <=> (*PTRANS*)
  OBJECT HUM0
  TO (*PROX* PART HUM1)
  INST ((ACTOR HUM0 <=> ($CAR)
        TIME (FORM26)))
 TIME (FORM27))
```

HUM2 is a token generated for "he". On the property list of HUM2 is the following information:

```
HUM2

 TOKEN:         T

 CLASS:         (*PERSON*)

 GENDER:        (*MASC*)

 REF:           (*DEF*)
```

That is, HUM2 refers to some unknown but previously mentioned male person. HUM1 contains a representation for the farmer that includes his location. Now PAM proceeds to explain this event.

```
NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
USE-VEHICLE-REQ

EXPLANATION IS
 PLAN: (*PB-USE-VEHICLE* PLANNER
HUM0 LOCATION (*PROX* PART HUM1))
```

PAM infers that John drove his car over to the farmer,

```
NO PREDICTION CONFIRMED

ASSUMING EXPLANATION
CONTINUING SEARCH
```

| | |
|---|---|
| TESTING EXPLANATION OFFERED BY<br>USE-VEHICLE-GOAL-EPISODE-REQ | |
| EXPLANATION IS GOAL:<br>(*DPROX* PLANNER HUMO OBJECT HUMO<br>LOCATION (*PROX* PART HUM1)) | that he did this because he wanted to be near the farmer, |
| PLAN: (*PB-USE-VEHICLE* PLANNER<br>HUMO LOCATION (*PROX* PART HUM1)) | |
| NO PREDICTION CONFIRMED | |
| ASSUMING EXPLANATION<br>CONTINUING SEARCH | |
| TESTING EXPLANATION OFFERED BY<br>USE-PERSON-REQ | |
| EXPLANATION IS | |
| NAMED-PLAN USE-PERSON RELATES<br>GOAL: (*DPROX* PLANNER HUMO OBJECT<br>HUMO LOCATION (*PROX* PART HUM1))<br>TO<br>GOAL: (*DKNOW* PLANNER HUMO<br>RECIPIENT HUMO FACT ((ACTOR HUMO IS<br>(*PROX* PART (*UNSPEC* CLASS<br>(*LOCATION*) ))) TIME (TIMK2))) | and that he wanted to be near the farmer in order to use the farmer for some purpose. |
| EXPLANATION CONFIRMS PREDICTION<br>SUITABLE-PLAN-REQ | A copy of SUITABLE-PLAN-REQ, activated during the processing of the previous sentence, finds that it can explain this inference. |
| FOUND EXPLANATION SEQUENCE: | |
| USE-VEHICLE-PLAN-REQ -><br>USE-VEHICLE-GOAL-EPISODE-REQ<br>-> USEPERSON-REQ | |

For SUITABLE-PLAN-REQ to explain John's wanting to be near the farmer, the condition of SUITABLE-PLAN-REQ must use a form of knowledge called a <u>Namedplan</u>. Namedplans relate goals by posing intermediate but unknown plans. Namedplans are necessary because PAM assumes that goals are instrumental to other goals through particular plans. For example, if John wants to find out something, and decides to ask somebody, then being near that person fulfills a precondition for the ASK planbox. However, it the current story, PAM was not yet told what plan John was going to use. Furthermore, being near the farmer can also be instrumental to other plans for this goal; John might try any of the plans in the PERSUADE package at this point.

Thus some goal states are often instrumental to many plans for a goal. A reader should be able to establish this instrumentality even if the particular plan has not yet been inferred. A Namedplan is an entity that is used to make this connection. For example, the Namedplan Use-person states that being near a person is often instrumental to using that person to find out something. When the condition of SUITABLE-PLAN-REQ examines a structure and finds that it contains an unknown plan, it tries to find a Namedplan that can relate the structure to the goal.

PAM maintains a list of Namedplans under each goal type. Each Namedplan is essentially a list of goals related by instrumentality. One of the Namedplans under Delta-know is Use-person. Use-person states that Delta-prox is often instrumental to this goal. The condition of SUITABLE-PLAN tests the structure it is focused at to see if it is a Delta-prox. In this case it is, so the request is triggered by the structure in its focus.

While SUITABLE-PLAN-REQ is making this determination, it checks to see that the planner of this plan is the same planner as the planner of the goal episode for which it thinks the plan is suitable. This check is necessary to be sure that both goals belongs to the same planner. The planner of the goal already in the story representation is HUM0, representing John, and the planner of the newly inferred goal is HUM2, representing the pronoun "he". SUITABLE-PLAN-REQ compares these tokens using a routine called AGREEWITH. AGREEWITH checks to see if one of its arguments is a pronoun, and if so, returns true if the pronoun has no properties not on the other token. HUM2 specifies only that it is male and human, so AGREEWITH returns a match. It also adds the pair (HUM2 HUM0) onto a reference list.

Before PAM begins the next phase, it takes the input conceptualization and replaces each occurrence of the first element of each pair on the reference list with its corresponding second element. In this case, all references to HUM2 are replaced by HUM0. Now when PAM changes the story representation in the next phase, only HUM0 will appear in the representation:

| | |
|---|---|
| *** ADDING TO STORY REPRESENTATION: | |
| PLAN: (*PB-USE-VEHICLE* PLANNER HUM0 LOCATION (*PROX* PART HUM1)) | PAM incorporates the inferences made explaining the previous sentence. |
| INFERRED GOAL: (*DPROX* PLANNER HUM0 OBJECT HUM0 LOCATION (*PROX* PART HUM1)) | |
| INPUT CONFIRMS PREDICTION SUITABLE-PLAN-REQ | |

```
NEXT INPUT IS:
 (HE ASKED HIM WHERE HE WAS)

CONCEPTUALIZATION IS:
((ACTOR HUM3 <=> (*MTRANS*)
  MOBJECT
  ((CON
   ((ACTOR HUM4 <=> (*MTRANS*)
     MOBJECT
      ((ACTOR HUM5 IS (*PROX*
        PART (*UNSPEC* CLASS
        (*LOCATION*))))
       TIME (FORM43))
     TO (*CP* PART HUM3))
    TIME (FORM44))
    LEADTO
     ((ACTOR HUM3 TOWARD
       (*JOY* VAL (NIL)))
      INC (2.) TIME (FORM45))))
  TO (*CP* PART HUM4))
 TIME (FORM46))
```

The representation of this sentence contains tokens for three pronouns, HUM3, HUM4, and HUM5, since PAM has not yet found referents for any of them.

```
NOT A PREDICTED INPUT

BEGIN SEARCH FOR EXPLANATION

TESTING EXPLANATION OFFERED BY
ASK-REQ
```

The input is not expected, so it is subjected to the discrimination net. PAM finds that it is an instance of the ASK planbox.

```
EXPLANATION IS
 PLAN: (*PB-ASK* PLANNER HUM0
PERSUADEE HUM1 REQUEST ((ACTOR HUM1
<=> (*MTRANS*) MOBJECT ((ACTOR HUM0
IS (*PROX* PART (*UNSPEC* CLASS
(*LOCATION*)))) TIME (TIMK7))
TO (*CP* PART HUM0)) TIME (TIMK6)))

***** ADDITIONAL INFORMATION FOUND
ABOUT PLAN FOR
 GOAL: (*DKNOW* PLANNER HUM0
RECIPIENT HUM0 FACT ((ACTOR HUM0 IS
(*PROX* PART (*UNSPEC* CLASS
(*LOCATION*)))) TIME (TIMK2)))
```

When SUITABLE-PLAN-REQ connected the "be near farmer" Goal with the "know one's location" goal, it did so by positing an unknown mediating plan. It also set up a request trying to fill in the gap for this plan. This request identifies the ASK plan as a plan for this goal.

```
FOUND EXPLANATION SEQUENCE:

ASK-REQ
```

```
*** ADDING TO STORY REPRESENTATION:

INFERRED
 PLAN: (*PB-ASK* PLANNER HUMO
PERSUADEE HUM1 REQUEST ((ACTOR HUM1
<=> (*MTRANS*) MOBJECT ((ACTOR HUMO
IS (*PROX* PART (*UNSPEC* CLASS
(*LOCATION*)))) TIME (TIMK7)) TO
(*CP* PART HUMO)) TIME (TIMK6)))
```

The information is added to the story representation.

```
LOADING PREDICTION
COMPLIANCE-GOAL-REQ

LOADING PREDICTION
REFUSAL-GOAL-REQ

FINISHED UNDERSTANDING PHASE
```

The request that built the ASK planbox structure also makes suggestions that look for compliance and refusal. However, since the story ends at this point, we will not get to see these requests in action. Instead, the following post-understanding tasks are performed:

---

```
*(TELL)

WHO SHOULD TELL THE STORY?  *JOHN
```

I wanted to get someplace, but I didn't know where I was. So I pulled over to a farmer standing nearby, and I asked him to tell me where I was.

```
WHO SHOULD TELL THE STORY?  *THE FARMER
```

I was standing by the roadside when a man pulled up in a car. He asked me to tell him where he was.

```
*(DOQA)
```

QUESTION: ?Q1
 Why did John pull over to a farmer?
 Because he wanted to know where he was.

QUESTION: ?Q2
 Why did John ask a farmer to tell him where he was?
 Because he wanted to get someplace.

QUESTION: ?Q3
 What were the consequences of John's pulling over to a farmer?
 This enabled him to ask him where John was.

QUESTION: ?Q4
What happened when John pulled over to a farmer?
He asked him to tell him where he was.

QUESTION: ?Q5
What happened when John asked a farmer to tell him where he was?
I don't know.

QUESTION: ?Q6
How did John go over to a farmer?
He drove over to him.

[PHOTO: Recording terminated Sat 5-Aug-78 2:49PM]

The following is part of the actual representation produced by PAM for this story. The expanded form of the representation (i. e., without nodes) is used wherever possible. Unexpanded nodes are generally backpointers to previously displayed structures. Listnodes in the representation have been replaced with lists of the form "(*LIST* CON1 CON2 CON3)", with "(*LIST*)" denoting an empty list.

```
CON186:
(*GOAL-EPISODE*
 GOAL-FORM ((GOAL (*DPROX* PLANNER HUMO
                   OBJECT HUMO LOCATION (NIL))
            RELATIONS (*LIST*)
            OUTCOME (NIL)
            SOURCE  (NIL)))
 ATTEMPTS (*LIST* CON165))

CON165:
((PLAN ((PLANBOX (*PB-USE-VEHICLE* PLANNER HUMO LOCATION (NIL))
                  OUTCOME (NIL) SOURCE (CON186)
         ACTIONS (*LIST*)
         SUBEPISODES (*LIST* CON184)))
  SOURCE (CON186)
  OUTCOME (NIL)))

CON184:
(*GOAL-EPISODE*
 GOAL-FORM ((GOAL (*DKNOW* PLANNER HUMO RECIPIENT HUMO
                   FACT ((ACTOR HUMO IS (*PROX* PART
                         (*UNSPEC* CLASS (*LOCATION*))))
            SOURCE (SUBGOAL PLAN CON165)
            RELATIONS (*LIST*)
            OUTCOME (NIL)))
 ATTEMPTS (*LIST* CON139))
```

```
CON139:
((PLAN
  ((PLANBOX
    (*PB-ASK* PLANNER HUM0 PERSUADEE HUM1
      REQUEST ((ACTOR HUM1 <=> (*MTRANS*)
                MOBJECT ((ACTOR HUM0 IS (*PROX* PART
                           (*UNSPEC* CLASS (*LOCATION*))))
                         TIME (TIMK7))
                TO (*CP* PART HUM0)) TIME (TIMK6))))
     ACTIONS (*LIST* CON444)
     SUBEPISODES (*LIST* CON334)))
  SOURCE CON184
  OUTCOME (NIL)))

CON444:
((ACTION
 ((ACTOR HUM0 <=> (*MTRANS*)
   MOBJECT ((CON
            ((ACTOR HUM1 <=> (*MTRANS*)
              MOBJECT
               ((ACTOR HUM0 IS (*PROX* PART (*UNSPEC* CLASS
                (*LOCATION*)))) TIME (TIMK7))
              TO (*CP* PART HUM0))
             TIME (TIMK6))
            LEADTO
            ((ACTOR HUM0 TOWARD (*JOY* VAL (NIL))) INC (2.)
             TIME (TIMK8))))
   TO (*CP* PART HUM1)) TIME (TIMK5))
  REASONS (*LIST* CON139)
  INITIATES (*LIST*)
  RESULTS (*LIST*)
  LASTEVENT CON369
  NEXTEVENT CON370))

CON334:
(*GOAL-EPISODE*
 GOAL-FORM ((GOAL (*DPROX* PLANNER HUM0 OBJECT HUM0
                            LOCATION (*PROX* PART HUM1))
            RELATIONS (*LIST*)
            OUTCOME (*SUCCEED*)
            SOURCE (CON139)))
 ATTEMPTS (*LIST* CON315))

CON315:
((PLAN ((PLANBOX (*PB-USE-VEHICLE* PLANNER HUM0
                  LOCATION (*PROX* PART HUM1))
         ACTIONS (*LIST* CON205)
         SUBEPISODES (*LIST*))
  SOURCE (CON334)
  OUTCOME (*SUCCEED*)))
```

```
CON205
((ACTION ((ACTOR HUMO <=> (*PTRANS*)
          OBJECT HUMO TO (*PROX* PART HUM1)
          INST ((ACTOR HUMO <=> ($CAR))
                TIME (TIMK4)))
         TIME (TIMK3))
  REASONS (*LIST* CON315)
  INITIATES (*LIST*)
  RESULTS (*LIST*)
  LASTEVENT CON203
  NEXTEVENT CON204))
```

# APPENDIX

## Planboxes

Planboxes are the basic building blocks out of which more complicated plans are constructed. In Chapter 3, I presented the planboxes postulated by Schank and Abelson. I have found it necessary to add to their list, and a summary of the planboxes I used is found below.

The planboxes I required are of three kinds: scripts used as planboxes, very simple planboxes, and "major" planboxes. Schank and Abelson make a strong distinction between a plan and a script which I found it useful to maintain. Within a script, it is not necessary to ask for explanations for an event. If we find an event in a script, we need not perform plan application to understand it.

However, in terms of fulfilling a goal, a script is just a stereotyped planbox. For example, if PAM learns that John is going to a restaurant, it infers that John may be hungry. The same inference is made by PAM if it learns that John trade his top to Bill for a candy bar. It is easier to recognize a stereotyped plan than a novel one, but the formal structure is the same.

Thus I felt free to add knowledge about scripts to PAM as simple planboxes. For example, a planbox for Satisfy-hunger is DO-$RESTAURANT, i. e., do the restaurant script. The preconditions of the plan are to have money and be at the restaurant. PAM was not given any knowledge of the content of this script. Thus PAM has no idea about what goes on inside a restaurant, but it can reason about why a person might go there. Another example of a script used as a plan is the marriage script, which is used for the goal of becoming married.

The second class of planboxes I needed might also be considered scripts, and simple ones at that. The planbox EAT is an example of one of these. EAT consists simply of executing the primitive action INGEST on an edible substance via the mouth. Such planboxes are not particularly interesting in their own right, and serve mostly a technical function. That is, PAM needs a place to record the fact that eating is a way to satisfy hunger, that a person must eat food, etc. The planbox EAT serves to organize this information.

Postulating the existence of such planboxes also helps PAM maintain a uniform syntax for its representations. In PAM's story representations, a goal can give rise to an action only through a plan, and one goal can be instrumental to another only if the first fulfills a precondition for a plans for the second. For example, having an apple is instrumental to satisfying hunger because it meets a precondition of EAT. Thus by postulating plans like EAT, the same format can be used to represent this situation as any other situation involving instrumental goals.

Some other very simple planboxes include

1. TAKE - to take possession of an object one already controls.

2. WRITE - to communicate by writing.

3. SPEAR - holding things together by skewering them.

4. SUPPORT - supporting an object by putting it on or in another object.

Finally, there are new planboxes that have greater generality. These include

1. SOCIAL-INTERACTION - This is actually a set of different script-like behavior, including making conversation, having sex, playing games, etc.

2. USE-CONTAINER - This plan is used to maintain a quality of an object by placing it inside of something. For example, using a thermos is an instance of this planbox where the quality is temperature and the substance a liquid. Packing a delicate instrument with foam is an example of using this planbox to maintain the physical condition of an object.

3. UNDO-PRECONDITION, AVOID-DETECTION, and DISTRACT are all anti-planboxes and were discussed in Chapter 9.

4. COMPLY is specific to avoiding threats. This was also discussed in Chapter 9.

The following is a list of all the planboxes actually used in PAM. Descriptions are given for planboxes not found in Schank and Ableson (1977).

D-CONT with the planner as recipient:

1. TAKE - Simply ATRANSing the object to oneself. Preconditions are

   1. No one else controls object.

   2. Planner is near object.

2. ASK

3. THREATEN

4. INFORM-REASON

5. EXPRESS-DESIRE
6. INVOKE-THEME
7. OVERPOWER
8. BARGAIN-OBJECT

D-CONT with the recipient other than the planner:

1. GIVE - ATRANS object to recipient. Precondition are
    1. Planner control object

D-KNOW with the planner being the recipient of the knowledge:

1. READ - ATTENDing to printed symbols. Preconditions are
    1. Ability to read.
    2. Control of printed material.
2. ASK
3. THREATEN
4. INFORM-REASON
5. EXPRESS-DESIRE
6. INVOKE-THEME
7. BARGAIN-OBJECT

D-KNOW with the planner being other than the recipient.

1. WRITE - To create printed symbols for someone else to READ. Preconditions are
    1. Ability to write.
    2. Control of paper, ink and pen type objects.

2. TELL - To verbally communicate something. Preconditions are

    1. Communication link (i. e., proximity or telephone, two-way radio, etc.).

D-PROX

1. WALK

2. USE-VEHICLE

S-HUNGER

1. EAT - INGEST some food. Requires

    1. Control of food.

2. DO-$RESTAURANT - The script version of this plan. Requirements are those of the restaurant script.

S-SLEEP

1. DO-$SLEEP - The sleeping script. Requirements are

    1. Control of comfortable surface.

    2. Low level of sensory stimulation (i. e., quiet and dark).

E-ENTERTAINMENT

1. WATCH - ATTEND to entertainment. Preconditions are

    1. Access to source of entertainment

E-COMPANY

1. SOCIAL-INTERACTION - A set of different script-like behavior, including making conversation, having sex, playing games, etc. In addition to the particular preconditions of each script, an agreeable companion is always required.

A-CONNECTION (connect together to physical objects)

1. SPEAR - PROPEL a skewering device into the objects. Preconditions are

    1. Control skewering device, objects.

    2. Skewering device must be capable of piercing objects.

2. SUPPORT - PTRANS one physical object on top of another. Preconditions are

    1. Control objects

    2. One object can support the other.

Preserve a quality of an object

1. USE-CONTAINER - PTRANS substance inside an appropriate container to maintain a quality. Preconditions are

    1. Control of container and substance.

    2. Substance fits into container.

Preserve some state someone is requesting you to change

1. REFUSE - MTRANS that you do not wish to comply with the request. Preconditions are

    1. Planner is request to perform an action.

Planboxes for anti-planning

1. UNDO-PRECONDITION - Undoing a precondition for an opponent's plan. Precondtions are

    1. Knowing opponent's plan.

2. AVOID-DETECTION - Performing a plan so as not to be detected. Preconditions are

    1. Opponent not attending to planner.

3. DISTRACT - Focusing opponents attention else while performing a plan.

4. COMPLY - Giving in to a threat. Preconditions are

    1. A threat was made.

    2. Ability to comply with the threat.

BIBLIOGRAPHY


1] Abelson, R. P. (1973). The structure of belief systems. In R. C. Schank and K. M. Colby (eds.), Computer Models of Thought and Language. Freeman, San Francisco.

2] Abelson, R. P. (1975). Concepts for representing mundane reality in plans. In D. Bobrow and A. Collins (eds.), Representation and Understanding: Studies in Cognitive Science. Academic Press, New York.

3] Bobrow, D. G. and Collins, A., editors. (1975). Representation and Understanding: Studies in Cognitive Science. Academic Press, New York.

4] Carbonell, J. G. Jr. (1978). POLITICS: Automated Ideological Reasoning. Cognitive Science Vol. 2, No. 1.

5] Carbonell, J. G. Jr. (1978). Ph. D. thesis, Yale University Department of Computer Science (in progress).

6] Charniak, E. (1972). Towards a model of children's story comprehension. AI TR-266, MIT.

7] Charniak, E. (1978). On the use of framed knowledge in language comprehension. To appear in Artificial Intelligence.

8] Cullingford, R. E. (1978). Script Application: Computer Understanding of newspaper stories. Yale University Research Report #116.

9] Feigl, H., and Maxwell, G., editors. (1962). Minnesota Studies in the Philosophy of Science. Vol III, University of Minnesota Press, Minneapolis, Minn.

10] Fikes, R. E. and Nilsson, N. J. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence, Vol. 2, No. 3-4, pp. 189-208.

11] Fodor, J. (1975). The Language of Thought. Thomas Y. Cromwell, New York.

12] Joshi, A. K. and Rosenschein, S. J. (1976). Some Problems of Inferencing: Relation of Inferencing to Decomposition of Predicates. In Proceedings of the International Conference on Computational Linguistics (COLING 1976), Ottawa, Canada.

13] Joshi, A. K. and Weischedel, R. (1977). Computation of a subclass of inferences: presupposition and entailment. American Journal of Computational Linguistics.

14] Katz, J., and Fodor, J. (1963). The structure of a semantic theory. Language 39.

15] Kuno, C. and Oettinger, A. (1962). Multiple path syntactic analyzer. In C. M. Popplewell (ed.), Information Processing 1962, North-Holland, Amsterdam.

16] Lehnert, W. G. (1978). Representing physical objects in memory. Yale University Research Report #131.

17] McCawley, J. D. (1968). The role of semantics in a grammar. In E. Bach and R. Harms, (eds.), Universals in Linguistic Theory. Holt, Rinehart and Winston, New York.

18] McDermott, D. V. (1974). Assimilation of new information by a natural language-understanding system. MIT AI Laboratory TR-291.

19] McDermott, D. V. (1977). Flexibility and efficiency in a computer program for designing circuits. MIT AI Laboratory TR-402.

20] Meehan, J. (1976). The Metanovel: Writing stories by computer. Yale University Research Report #74.

21] Minsky, M. (1974). A framework for representing knowledge. MIT. AI Memo No. 306.

22] Miller. G. A. and Johnson-Laird P. (1976). Language and Perception. Harvard University Press, Cambridge, Mass.

23] Nelson, K. (1974). Concept, word and sentence: Interrelations in acquisition and development. Psychological Review, Vol. 81, No. 4, 267-285.

24] Newell A., Shaw J. C., and Simon, H. A. (1959). Report on a general problem-solving program. In Proceedings of the International Conference on Information Processing (ICIP), Paris, UNESCO House.

25] Newell A., and Simon, H. A. (1972). Human Problem Solving. Prentice-Hall, Inc., Englwood Cliffs, New Jersey.

26] Norman, D. A. and Rumelhart, D. E. (1975). Explorations in Cognition. W. H. Freeman and Co., San Francicso.

27] Polti, G. (1916). The Thirty-Six Dramatic Situations. The Editor Company, Ridgewood, New Jersey.

28] Propp, V. (1968). Morphology of the Folktale. University of Texas Press, Austin.

29] Rieger, C. (1975). Conceptual memory. In R. C. Schank, (ed.), Conceptual Information Processing. North-Holland, Amsterdam.

30] Rieger, C. (1975a). The commonsense algorithm as a basis for computer models of human memeory, inference, belief, and contextual language comprehension. In R. Schank and B. Webber (eds.). Proceedings of the Conference on Theoretical Issues in Natural Language Processing. Distributed by the Association for Computational Linguistics.

31] Riesbeck, C.K. (1975). Conceptual analysis. In R. C. Schank (ed.), Conceptual Information Processing. NorthHolland, Amsterdam.

32] Riesbeck, C.K. and Schank, R. C. (1976). Comprehension by computer: Expectation-based analysis of sentences in context. Yale University Research Report #78.

33] Rumelhart, D. E. (1976). Understanding and Summarizing brief stories. Center for Human Information Processing Technical Report No. 58. University of California, San Diego.

34] Sacerdoti, E. D. (1974). Planning in a Hierarchy of Abstraction Spaces. Artificial Intelligence, Vol. 5, No. 2, pp. 115-135.

35] Sacerdoti, E. D. (1977). A Structure for Plans and Behavior. Elsevier North-Holland, Amsterdam.

36] Salmon, W. C., Jeffrey, R. C., and Greeno, J. G. (1971). Statistical Explanation and Statistical Relevance. University of Pittsburg Press. Pittsburg, Pennsylvania.

37] Schank, R. C. (1974). Understanding paragraphs. Technical Report #6, Istituto per gli Studi Semantici e Cognitivi, Castagnola, Switzerland.

38] Schank, R. C. (1973). Causality and reasoning. Technical Report #1, Istituto per gli Studi Semantici e Cognitivi, Castagnola, Switzerland.

39] Schank, R. C. (1975). Conceptual Information Processing. North-Holland, Amsterdam.

40] Schank, R. C. (1975a). The role of memory in language processing. In C. Cofer and R. Atkinson, (eds.). The Nature of Human memory. Freeman, San Francisco.

41] Schank, R. C. (1975b). The structure of episodes in memory. In D. Bobrow and A. Collins (eds.), Representation and Understanding: Studies in Cognitive Science. Academic Press, New York.

42] Schank, R. C. and Abelson, R. P. (1977). Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Press, Hillsdale, N.J.

43] Schank, R. C. Goldman, N., Reiger, C., and Riesbeck, C. (1973). MARGIE: Memory, analysis, response generation and inference in English. In Proceedings of the Third International Joint Conference on Artificial Intelligence.

44] Schank, R. C., and Rieger, C. (1974). Inference and the computer understanding of natural language. Artificial Intelligence. 5(1974), 373-412.

45] Schank, R. C., and Webber B., editors. (1975). Proceedings of the Conference on Theoretical Issues in Natural Language Processing. Distributed by the Association for Computational Linguistics.

46] Schank, R. C., Wilensky, R., Carbonell, J. G. Jr., Kolodner, J. L., Hendler, J. A. (1978). Representing Attitudes: Some Primitive States. Yale University Research Report #128.

47] Schank, R. C. and Yale A. I. Project (1975). SAM -- A story understander. Yale University Research Report #43.

48] Schmidt, C. F., et. al. (1976). Recognizing plans and summarizing actions. Proceedings of the Conference on Artificial Intelligence and the Simulation of Behavior. pp. 291-306. Edinburgh, Scotland.

49] Schmidt, C. F., Sridharan, N. S., and Goodson, J. L. (1978). The plan recognition problem: An intersection of psychology and artificial intelligence. To appear in Artificial Intelligence, special issue on applications in the sciences and medicine, 1978.

50] Selfridge, O. G. (1959). Pandemonium: A Paradigm for Learning. Proceedings of the Symposium on Mechanisation of Thought Processess. 2 vols. National Physical Laboratory, Teddington, England. London: H.M. Stationary Office, 1959, pp. 511-529.

51] Sussman, G. J. (1975). A Computer Model of Skill Acquisition. American Elsevier, New York.

52] Waterman and Hayes-Roth, editors (1978). Pattern-directed Inference Systems. Academic Press, New York.

53] Weinreich, U. (1970). Explorations in semantic theory. In D. D. Steinberg and L. A. Jakobovits, (eds.). Semantics. Cambridge Univ. Press, London.

54] Wilks, Y. (1975). A preferential pattern-seeking, semantics for natural language inference. Artificial Intelligence 6, 53-74.

55] Wilks, Y. (1977). Knowledge Structures and Language Boundaries. In R. Schank and B. Webber (eds.) Proceedings of the Conference on Theoretical Issues in Natural Language Processing. Distributed by the Association for Computational Linguistics.

56] Wish, M. (1975). Comparisons among multidimensional structures of interpersonal relations. Unpublished manuscript, Bell Laboratories, Murray Hill, New Jersey.

57] Wish, M. (1976). Perceived dimensions of interpersonal relations. _Journal of Personality and Social Psychology._